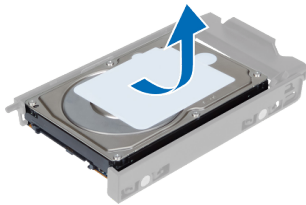


Same PC, new HD

Serge Y. Stroobandt

Copyright 2016, licensed under [Creative Commons BY-NC-SA](#)

Introduction



Here are my notes about how to change the system hard disk on your GNU/Linux computer. This comes handy when you want to upgrade your «*spinning rust*» boot drive to a much quicker **solid-state drive (SSD)**.

Required hardware



With the PC **powered down**, install and connect the new hard drive. For the time being, **the old hard drive remains in place** to allow the copying of its partitions to the new drive.

However, you may be dealing with a laptop computer, and/or have only one hard drive bay or **SATA port** available. If this is the case, use **a SATA to USB adaptor** to temporarily connect the new hard drive.



Figure 1: SATA to USB3 adapter

Use a live DVD or USB



Instead of meddling with `chroot`, I opted for the more straightforward technique of booting from either a live DVD or a live USB.

At start-up, you most probably will have to instruct your computer's BIOS or (more modern) UEFI to boot from the live DVD or USB instead of any HD. The BIOS or UEFI can be accessed by hitting a specific `key` during start-up. Which `key`, is briefly posted during the boot sequence. It differs from computer to computer.

GParted «copy party»

Creating a partition table

If you have a brand new hard drive, chances are it will come without any partition table. There are two choices:

- Either a 1983-era MS-DOS style Master Boot Record (MBR),
- Or a more modern GUID Partition Table (GPT).

Obviously, GPT offers many advantages over MBR. If you want to boot in UEFI-mode, it is even your only choice. However, if you would like to dual boot or when you are using an older BIOS without GPT support (like I do), you are basically stuck to MBR. Speed wise there is no difference between both technologies. When decided, proceed with GParted to Device → Create Partition Table...

Copying partitions with GParted



With the live session running, carefully identify the old and new hard drive using the `lsblk` command.

```
$ sudo apt install util-linux
$ lsblk
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
sda   8:0    0 279.5G 0 disk
├─sda1 8:1    0    1K 0 part
├─sda5 8:5    0   512M 0 part /boot
├─sda6 8:6    0    4.5G 0 part [SWAP]
├─sda7 8:7    0    64G 0 part /
└─sda8 8:8    0 210.5G 0 part /home
sdb   8:80   0 298.1G 0 disk
sr0   11:0   1  1024M 0 rom
```

From the Settings menu, launch **GParted**. A cool feature of GParted is that it allows for copying partitions from one drive to another. This works for all partitions, except for the [SWAP] partition which will be in use by the live session.

Creating swap... or not

Anyhow, the current swap contents is of no importance to us. This leaves us with two options:

- Either use GParted to create an equally sized swap file system on the new hard disk, which is easy enough,
- Or, skip the creation of a swap partition all together.

This calls for an explanation. All depends on the underlying hardware:

- On a laptop that should be able to hibernate, create a swap partition slightly larger than the available RAM.
- On a desktop computer with an ordinary «spinning rust» hard disk, it does not harm to create a swap partition; especially if you plan to do memory intensive jobs like video editing.
- On a desktop computer with a new solid-state drive (SSD), it is better not to create any swap partition. This will reduce the wear of frequent writes and eventually conserve the life span of your SSD.

Resize partitions before copying to a smaller hard drive

Partitions copied to the new drive can be made bigger, but not smaller. If you are copying partitions to a smaller sized SSD, you will need to reduce the existing partitions on the old drive first.

After copying the partitions with GParted, the list of block devices may look similar to this. On the old sda drive, I had implemented logical volume management (LVM); on the new sdb drive I chose not to do so any more.

```
$ lsblk
NAME MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda   8:0    0 279.5G  0 disk
├─sda1 8:1    0    1K  0 part
├─sda5 8:5    0   512M  0 part /boot
├─sda6 8:6    0    4.5G  0 part [SWAP]
├─sda7 8:7    0    64G  0 part /
└─sda8 8:8    0 210.5G  0 part /home
sdb   8:80   0 298.1G  0 disk
├─sdb1 8:81   0   512M  0 part /boot
├─sdb2 8:82   0    4.5G  0 part
├─sdb3 8:83   0    44G  0 part /
└─sdb4 8:84   0 249.1G  0 part /home
sr0   11:0    1 1024M  0 rom
```

Reinstalling GRUB



Now, mount the partition of the new drive which would normally be used for booting (here `/dev/sdb1`).

```
$ sudo mount /dev/sdb1 /mnt
```

Finally, reinstall GRUB on the new drive:

```
$ sudo grub-install --boot-directory=/mnt /dev/sdb
```

The computer may now be rebooted from the new hard drive. Again, you might have to instruct your BIOS/UEFI to do so.

Edit `/etc/fstab` for speed

After performing all of the above, you reboot your computer ... only to discover that the start-up sequence of your computer is taking longer! What is happening? Remember the swap partition we either deleted or recreated? This results in the system wasting time during start-up trying to locate a swap partition by a Universally Unique Identifier (UUID) which no longer exists. As a matter of fact, hitting the `[F1]` key whilst the start-up splash screen is shown, will reveal the following message:

```
A start job is running for dev-disk-by ... 1min30
```

Luckily, this can be easily remedied after the first reboot, by editing `/etc/fstab`:

```
$ sudo nano /etc/fstab
```

- If you are no longer using any swap partition, simply delete the line in `fstab` containing the word `swap`.
- Should you have recreated the swap partition, you will have to change the UUID on the line in `fstab` containing the word `swap`.

To obtain the UUID of the newly created swap partition, it is rather convenient to use the `blkid` tool from the `util-linux` package.

```
$ sudo apt install util-linux
$ blkid /dev/sda2
/dev/sda2: UUID="438b7a73-11c5-466c-b795-fcf6d0997056" TYPE="swap"
PARTUUID="00035ac0-02"
```

There is also the «old-school» method for obtaining the UUIDs of all disk partitions:

```
$ ls -l /dev/disk/by-uuid
total 0
lrwxrwxrwx 1 root root 10 Sep  1 13:08 438b7a73-11c5-466c-b795-fcf6d0997056
-> ../../sda2
lrwxrwxrwx 1 root root 10 Sep  1 13:08 7c581345-6850-44df-a387-cdf62cb09bba
-> ../../sda4
lrwxrwxrwx 1 root root 10 Sep  1 13:08 8ef04775-ec47-4663-bc77-7e6f8980e195
-> ../../sda3
lrwxrwxrwx 1 root root 10 Sep  1 13:08 9b5c2cd5-43de-4a56-9da9-741710d28dc3
-> ../../sda1
```

Getting rid of a FlexNet warning

Occasionally, GRUB may reinstall on a used drive, but not without issuing the following warning:

```
/usr/sbin/grub-install:
warning: Sector 32 is already in use by the program `FlexNet`;
avoiding it.
This software may cause boot or other problems in future.
Please ask its authors not to store data in the boot track.
```

As usual, **you are not the only one** dealing with this issue on the Internet. In my case, the replacement drive happened to be a **decommissioned drive that previously held commercially licensed software under Microsoft Windows™**. *FlexNet* appears to be a digital contraption that keeps record of one's licenses by writing data to a sector in the **extended boot record (EBR)**. This is certainly not the kind of digital parasite we would like to see living in our EBR!

Removing FlexNet

Still within the same live DVD/USB session or in a new one, remove *FlexNet* from the affected drive (here `/dev/sdb`), using the sector number reported by **GRUB** (here sector 32):

```
$ sudo dd if=/dev/zero of=/dev/sdb bs=512 count=1 seek=32
```

There is no need to backup this usually empty part of the **extended boot record (EBR)**.

Reinstalling GRUB after FlexNet removal



Remaining within this session, if not earlier, mount the partition affected drive which normally serves for booting (here `/dev/sdb1`).

```
$ sudo mount /dev/sdb1 /mnt
```

Finally, reinstall GRUB on the affected drive with:

```
$ sudo grub-install --boot-directory=/mnt /dev/sdb
```

GRUB should now install without any warnings. You are now safe to reboot from the disinfected hard drive.



This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](#).
Other licensing available on request.

Unattended [CSS](#) typesetting with .

This work is published at http://hamwaves.com/same_pc.new_hd/en/.

Last update: Thursday, September 1, 2016.