

Moving Debian “Wheezy” and grub2 to a New Drive

Brian Dunn — BD Tech Concepts LLC

May 2014

Copyright © 2014 Brian Dunn

Abstract

How to move a Debian Linux “Wheezy” installation to a new hard-disk drive, without a total re-installation.

1 Introduction

This procedure is for moving a Debian Linux installation which is currently residing on an obsolete or gradually failing hard-drive, over to a fresh new drive. This is not an upgrade to a new version of the operating system, nor is it new install from scratch.

Special care must be taken during the preparation of the new drive so that it will function properly when it becomes the master boot drive. This process involves the use of “super-user” mode, and mistakes can cause the system to become un-usable.

Users of laptop computers or users who are not comfortable with super-user command-line actions might do well to simply copy all data to an external drive,

swap out the boot drive, and create a new installation from scratch, rather than try to copy everything in place as described below.

For a laptop, some way must be available to copy to a new laptop harddrive while the old drive is still installed. If this is not possible, a fresh install will be required.

Installations with multiple users will be affected by the use of single-user mode during much of the following. Inform all users that the system will be going off-line before continuing.

Single-user mode might disable access to the Internet, so it is useful to have access to a second computer during this process, to provide access to research or assistance should problems arise.

2 Back-up the old system

Data-loss or temporary in-accessability may occur during this process. Back-up information will be very important to have in the event that your boot drive becomes corrupted by any of the following steps.



WARNING:

For multi-user systems, it would be best to move to single-user mode before beginning the backups, so that data is not changed during the backups.

While it will be possible to access the Internet and use the computer for other purposes while performing the time-consuming backup and copy operations, it is best not to do anything important until the entire procedure is finished. Data changes, new downloads, new bookmarks, and other information may not appear on the back-ups or on the new boot drive. If in doubt, simply boot into single-user “recovery” mode to avoid problems. Otherwise, continue to use the regular operating mode, but realize that new activity or data may not survive the following procedures.

Be sure to back-up at least the following items, which usually hold your own data and related information and which would not be replaced by a fresh install.

- home directories

- /ext
- /var
- /usr
- databases

3 Install the new drive

1. Shut down the computer.
2. Remove power to the computer.
 - If possible, turn off the switch on the power supply but keep it plugged into the wall outlet. This removes power from the keyboard but keeps the computer grounded.
 - If there is no separate power supply switch, unplug the entire computer from the wall power supply.
3. While installing the new drive, be sure to touch the chassis of the computer on occasion to discharge static electricity. If the computer case is unplugged from the wall, and therefore not grounded, then use a ground strap connected to the center screw of a wall outlet plate.
4. While installing the new drive, try not to touch any exposed electronics or connectors.
5. Be sure to entirely install the new drive, including screwing it into the case chassis and closing the case so that proper airflow is available. This help to keep the drives cool while copying large amounts of data.
6. Keep the old drive installed as well. You will have to copy from one to the other.
7. Once the case is closed, turn on and/or plug in the power supply again.

4 Boot into single-user “recovery” mode

Reboot the computer and select single-user “recovery” mode. This prevents any changes from occurring in the background while moving information to the new drive.

The text-only recovery mode will not have access to the usual graphic interface, the Internet, etc.

5 Print out the current setup

These commands print information about how the drive partitions are currently set up.

```
lp /etc/fstab
df -h | lp
sudo blkid | lp
```

6 Plan the partitions on the new drive

Many systems are installed with just root and swap partitions, but there are security and stability advantages to using a separate partition for each of boot, usr, tmp, var, and home. The below instructions assume that this has already been done, so some adaptation will be required if you are moving from an old drive with only a few partitions to a new drive with many. Dual-boot systems may require extra care as well. Use the following as a general guide, and adapt as necessary.

The following is an example for a new drive which is more than 100 Gigabytes in size.

Table 1: Sample Partition Table

Partition	Type	Size	Flags	Label
boot	ext2	300 M	boot	myboot
swap	swap	(RAM size)		myswap
root	ext4	20 G		myroot
extended		(remaining)		
usr	ext4	50 G		myusr
tmp	ext2	10 G		mytmp
var	ext4	20 G		myvar
home	ext4	(remaining)		myhome

7 Create partitions on the new drive

7.1 Software

Partitions are created using a special software package. To see what is available, try:

```
man -k partition
```

KDE Partition Manager is an easy-to-use higher level package which might also be available.

7.2 Create the partitions

Create the partitions on the new drive.

- Assign a label for each partition. These must be different than the labels on the old drive.
- Be sure to set the boot flag on the boot partition.
- Be sure to select linux swap for the swap partition.
- Be sure to select ext2 or ext4 for non-swap partitions. See the above table.

If not using KDE Partition Manager, you may need the following software to assign labels to each partition:

```
tune2fs -L <label> <device>  
swaponlabel -L <label> <device>
```

8 Create mount directories

On the old/existing drive:

1. Create a new directory to hold the new mount points:

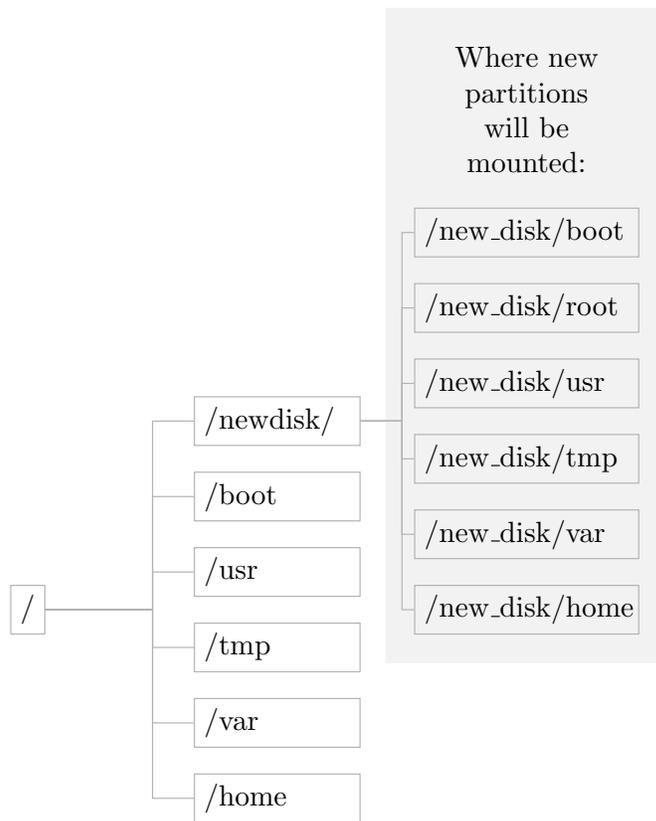
```
mkdir /new_disk/
```

2. Create the mount points for the partitions of the new drive:

```
cd /new_disk/  
mkdir /new_disk/boot/  
mkdir /new_disk/root/  
mkdir /new_disk/usr/  
mkdir /new_disk/tmp/  
mkdir /new_disk/var/  
mkdir /new_disk/home/
```

3. The resulting directory structure is shown in Figure 1 — “Directory Structure on the Old Drive”.

Figure 1: Directory Structure on the Old Drive



9 Mount the new partitions

Mount all the new drive's partitions.

i NOTE:

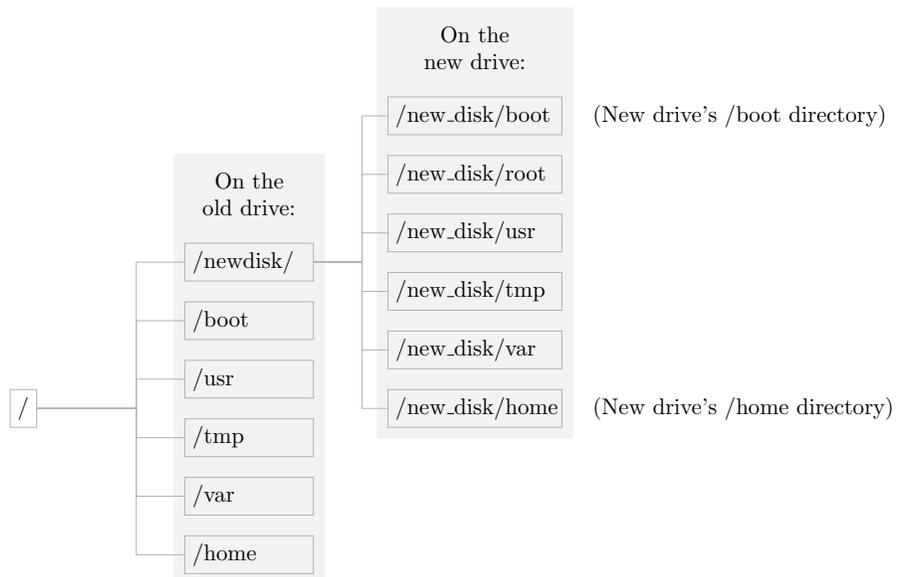
Use the command `blkid` to see whether the new drive is `/sda` or `/sdb`, etc.

Here, you want to use the device for the new drive. I.e. you will use `sdb7` instead of `sda7` if the above `blkid` command said that the new drive is `/sdb`.

```
cd /new_disk/  
mount -t ext2 /dev/sda2 /new_disk/boot/  
mount -t ext4 /dev/sda3 /new_disk/root/  
mount -t ext4 /dev/sda5 /new_disk/usr/  
mount -t ext2 /dev/sda6 /new_disk/tmp/  
mount -t ext4 /dev/sda7 /new_disk/var/  
mount -t ext4 /dev/sda8 /new_disk/usr/
```

After the partitions are mounted, the directory structure is as shown in Figure 2 — “Directory Structure After New Drive is Mounted”. `/home` will access the existing home directories on the old drive, and `/new_disk/home` will access the (currently empty) home directories on the new drive.

Figure 2: Directory Structure After New Drive is Mounted



10 Set permissions, owners, and groups of the new partitions

Partitions have their own set of permissions / owner / group.

1. View the desired settings. Look for the settings for /boot, /usr, etc. of the old drive.

```
ls -lh /      % This is for the old drive.
```

2. View current settings for the new partitions on the new drive:

```
ls -lh /new_disk/      % This is for the new drive.
```

3. Set special permissions in /new_disk/* to match the existing permissions in /

```
cd /new_disk/  
chmod a+w /new_disk/tmp  
chmod o+t /new_disk/tmp  
chmod a+w /new_disk/var  
chmod g+ws /new_disk/home  
chown :staff /new_disk/home
```

11 Create mount points inside the new drive

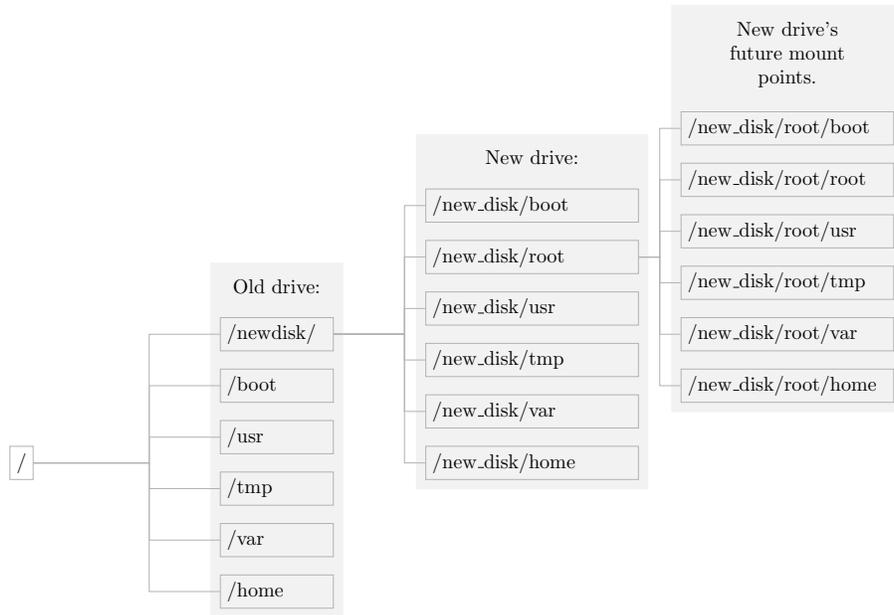
This is where the new drive will mount its own partitions.

```
cd /new_disk/root/      % Inside the new drive's root:  
mkdir /new_disk/root/boot/ % make new directories for use  
mkdir /new_disk/root/root/ % when the system is rebooted.  
mkdir /new_disk/root/usr/  
mkdir /new_disk/root/tmp/  
mkdir /new_disk/root/var/  
mkdir /new_disk/root/home/
```

Note that the permissions of the new partitions will override the permissions of these folders, so no special permission settings are needed for these folders.

The directory structure is now as shown in Figure 3 — “Directory Structure With Future Mount Points”. The mount points are empty directories, which will be filled when the computer is rebooted with the new drive’s Linux system.

Figure 3: Directory Structure With Future Mount Points



12 Copy to the new drive

12.1 Set single user mode

Reboot into single user "recovery" mode. This helps provide a stable system during the file copies.

i NOTE:

Access to the Internet may no longer be available in single-user mode. Plan on having another computer nearby to help look-up any issues which may arise.

12.2 Re-mount the new partitions

NOTE: use blkid to see whether the new drive is /sda or /sdb.

Here, you want to use the device for the new drive.

```
cd /new_disk/  
% Mount the partitions of the new disk:  
mount -t ext2 /dev/sda2 /new_disk/boot/  
mount -t ext4 /dev/sda3 /new_disk/root/  
mount -t ext4 /dev/sda5 /new_disk/usr/  
mount -t ext2 /dev/sda6 /new_disk/tmp/  
mount -t ext4 /dev/sda7 /new_disk/var/  
mount -t ext4 /dev/sda8 /new_disk/home/
```

12.3 Copy from old drive partitions to new drive partitions

Only copy things which will be actual partitions on the new drive.

Do NOT copy things which will be on the new drive's root as regular directories. Regular directories which are not their own partitions will be copied to the new root directory in the next step.

In the following, each of these is an actual partition in the new drive, not just a

directory:

```
cp -a /boot/* /new_disk/boot/  
cp -a /usr/* /new_disk/usr/  
cp -a /var/* /new_disk/var/  
cp -a /home/* /new_disk/home/
```

root will be handled next. Nothing needs to be done for tmp.

12.4 Special treatment for the new drive's root partition

Copy entire directories from / to /new_disk/root/ These are things which will NOT be their own partitions in the new drive.

Do NOT copy anything which is its own partition into /new/disk/root. In our examples, this means that you do not copy /boot, /usr, /var, /tmp, or /home, but you do copy the following directories:

```
cp -a /bin /new_disk/root/  
cp -a /dev /new_disk/root/  
cp -a /disks /new_disk/root/  
cp -a /etc /new_disk/root/  
cp -a /lib /new_disk/root/  
cp -a /media /new_disk/root/  
cp -a /sbin /new_disk/root/  
cp -a initrd* /new_disk/root/  
cp -a vmlinuz* /new_disk/root
```

12.5 Special directories

For proc, sys, mnt, and run, simply create new directories but do NOT copy anything into them. These are empty at boot.

```
mkdir /new_disk/root/proc  
mkdir /new_disk/root/sys  
mkdir /new_disk/root/mnt  
mkdir /new_disk/root/run
```

Set special permissions for proc:

```
chmod a-w /new_disk/root/proc
```

12.6 Copy any last files or directories

Look at the listings from the old directory for anything left-over.

Copy any other directories or files of interest from the old into the new root folder.

13 Adjust /etc/fstab on the new drive

Change each line to use the new partition labels:

Edit file /new_disk/root/etc/fstab

```
LABEL=myroot /      ext4 defaults,errors=remount-ro 0 1
LABEL=myswap none    swap  sw   0 0
LABEL=myboot  /boot  ext2  rw   0 2
LABEL=myhome  /home  ext4  rw   0 2
```

- Be sure that there is an entry for each of the new drive's partitions.
- Be sure that the labels are for the new drive.

14 Install grub2 on the new device

Grub2 must be installed while it thinks that the root user is on the new drive. This requires tricking it by using a “chroot” command.

14.1 Chroot to the new drive

```
mount -o bind /dev /new_disk/root/dev
mount -t proc none /new_disk/root/proc
mount -t sysfs none /new_disk/root/sys
cd /new_disk/root/
sudo chroot .
```

At this point, the command line is operating as if it is the root user, but with the new drive instead of the old drive.

14.2 Additional mounts for the new drive

It may be necessary to mount the new partitions into the new chroot:

```
% See if the /usr directory has already been mounted:
ls /usr

% If the directory looks empty and has not yet been mounted:
mount -t ext4 /dev/sda5 /usr
```

Do the same for other partitions which grub2 may use.

14.3 Update and install grub on the new drive

```
update-grub

% modify to match your new drive: sda, sdb, etc:
grub-install /dev/sda
```

At this point, both the old and also the new drive's boot menus will include options for booting from either the old drive or the new drive.

14.4 Unmount partitions

Unmount any additional partitions which were mounted in subsection 14.2 — “Additional mounts for the new drive” above.

```
umount /dev/sda5
% etc.
```

14.5 Exit the new root user

Exit the new chroot, and return to the old drive:

```
exit
```

15 Reboot, and set the boot drive in BIOS

1. Reboot the machine. As it reboots, try to enter the BIOS mode, as described below.

```
shutdown -r now    % causes the machine to reboot
```

2. While rebooting, press the DEL key multiple times to enter the BIOS setup. If this process does not bring you to the BIOS menu, consult your computer's manuals to find the correct procedure for doing so.
3. Once in the BIOS menu, find the setting for which drive is used to boot.
4. Select the new drive as the boot drive.
5. Tell the BIOS to save the changes and continue booting.

16 Boot into the new system

Grub2 probably recognized and lists the linux system on both the old and the new drives. Try to select the new drive. Once it boots up, verify that the new drive is being used by looking at `/etc/fstab` for your new drive's labels.

17 Verify that everything is working

If strange things are happening:

- Remember that a partition has its own permissions and ownership, which override the permissions and ownership of the mount point directory. See section 10 — “Set permissions, owners, and groups of the new partitions”, except that the new partitions are now mounted at `/boot`, etc instead of `/new_drive/boot`, etc.
- Be sure that the permissions are set correctly for the `proc` directory. See subsection 12.5 — “Special directories”.
- For the permissions of other directories, compare the old v.s. the new drives.

The old installation is still available by booting into the old drive, but remember that any changes will not be seen by the new drive.

18 Remove or repurpose the old drive.

After a while, when everything seems to be working properly, the old drive may be removed or reformatted for a new purpose.

Once the old drive is removed from the computer:

```
su
update-grub
```

This removes the reference to the old drive from the boot menu.