How to Dual-Boot OS X and Ubuntu

Nailen Matschke - nailen@caltech.edu

10/3/2015

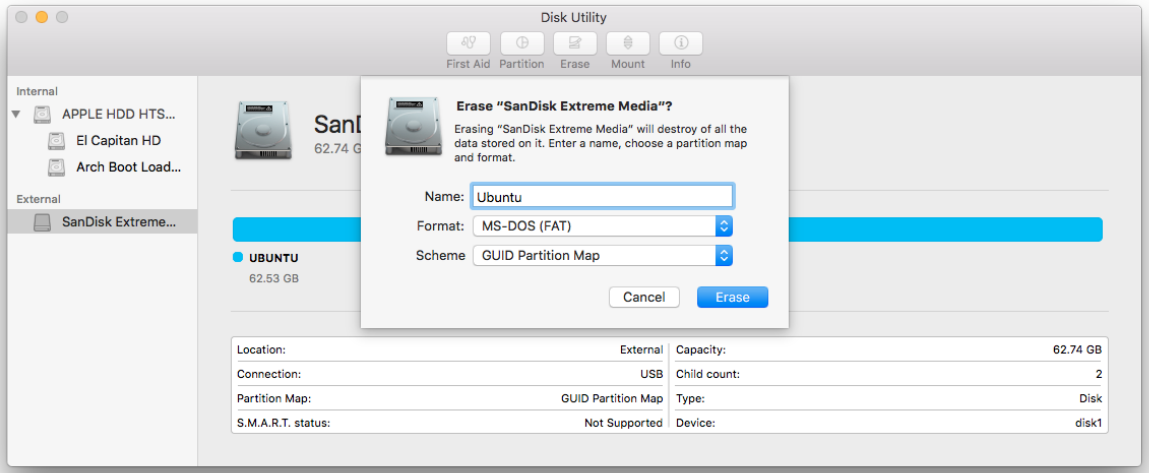
What you need:

1. A USB drive with at least 2 GB of space, that you don't mind wiping
2. A copy of Ubuntu (available here), or an Ubuntu-based Linux distribution (e.g. Mint)
3. A Mac running OS X, with at least 50 GB of unused disk space

In this guide, I'll be installing Ubuntu 15.04 on my MacBook Pro running OS X El Capitan, version 10.11.1 Beta.

Part 1: Creating a bootable USB drive

When adding a new OS, I like to start by creating the installation media, since it's the step least likely to do any damage to the computer. First, you'll want to plug your flash drive into your computer, and open up Disk Utility (in /Applications/Utilities/). Make sure that it's listed in Disk Utility's sidebar, probably under the "External" category. Now, select the drive so that its information appears in the main window, and then click on "Erase" at the top of the window. This will give you a drop down menu with options for how you want to format the drive, in which you should give it a memorable name, and change the "Format" field to "MS-DOS (FAT)," like so:



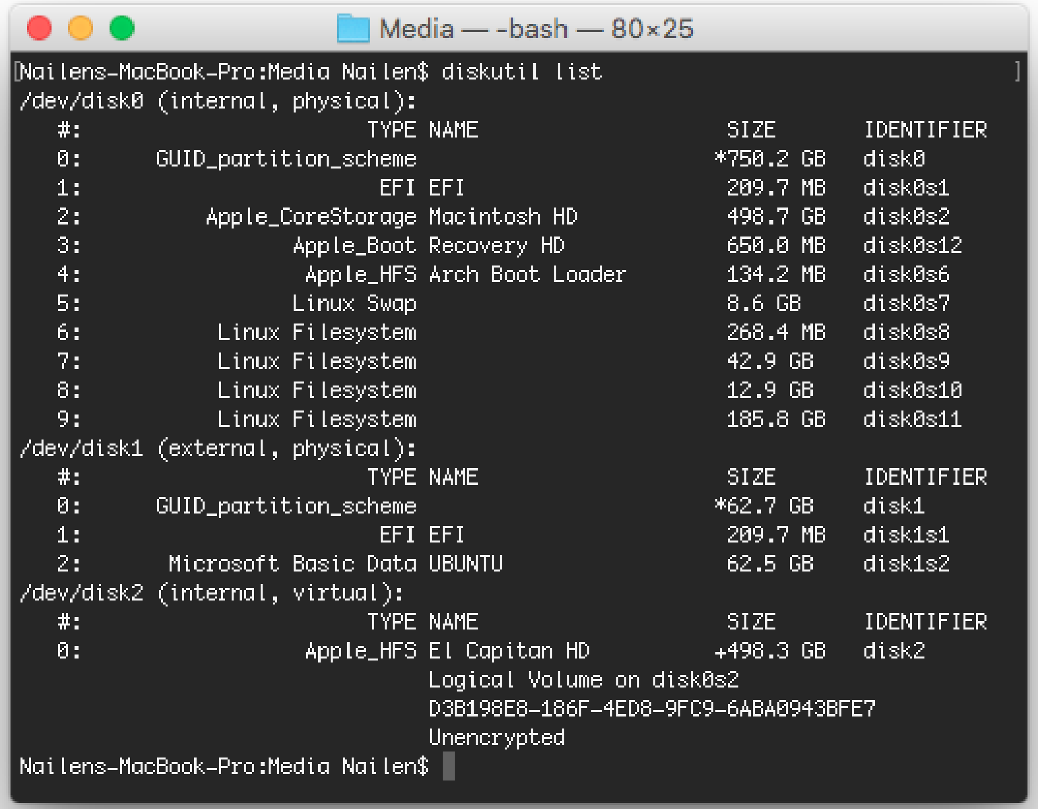
Erase "SanDisk Extreme Media"?

Erasing "SanDisk Extreme Media" will destroy of all the data stored on it. Enter a name, choose a partition map and format.

Name: Ubuntu
Format: MS-DOS (FAT)
Scheme: GUID Partition Map

Location:	External	Capacity:	62.74 GB
Connection:	USB	Child count:	2
Partition Map:	GUID Partition Map	Type:	Disk
S.M.A.R.T. status:	Not Supported	Device:	disk1

You can leave the "Scheme" field untouched. Hit "Erase," and the drive should be wiped and formatted within a few seconds. You can check that the process worked by opening up Terminal (also in /Applications/Utilities/) and running the `diskutil list` command, which will print out a list of your drives and their partitions, like this:



```
Nailens-MacBook-Pro:Media Nailen$ diskutil list
/dev/disk0 (internal, physical):
   #:                       TYPE NAME                    SIZE       IDENTIFIER
   0:      GUID_partition_scheme                        *750.2 GB   disk0
   1:                        EFI EFI                     209.7 MB   disk0s1
   2:          Apple_CoreStorage Macintosh HD            498.7 GB   disk0s2
   3:                 Apple_Boot Recovery HD             650.0 MB   disk0s12
   4:                  Apple_HFS Arch Boot Loader        134.2 MB   disk0s6
   5:                 Linux Swap                         8.6 GB     disk0s7
   6:           Linux Filesystem                         268.4 MB   disk0s8
   7:           Linux Filesystem                         42.9 GB    disk0s9
   8:           Linux Filesystem                         12.9 GB    disk0s10
   9:           Linux Filesystem                         185.8 GB   disk0s11
/dev/disk1 (external, physical):
   #:                       TYPE NAME                    SIZE       IDENTIFIER
   0:      GUID_partition_scheme                        *62.7 GB    disk1
   1:                        EFI EFI                     209.7 MB   disk1s1
   2:       Microsoft Basic Data UBUNTU                  62.5 GB    disk1s2
/dev/disk2 (internal, virtual):
   #:                       TYPE NAME                    SIZE       IDENTIFIER
   0:                  Apple_HFS El Capitan HD          +498.3 GB   disk2
                                 Logical Volume on disk0s2
                                 D3B198E8-186F-4ED8-9FC9-6ABA0943BFE7
                                 Unencrypted
Nailens-MacBook-Pro:Media Nailen$
```

The above image tells me that the USB drive is accessible at the path /dev/disk1, which has the correct partition scheme, format, and label. Disk Utility should also say the drive's path, in the "Device" field.

Your drive was probably automatically mounted by Finder after it was created, so next you want to unmount it via Terminal using the command:

```
diskutil unmountDisk /dev/diskN
```

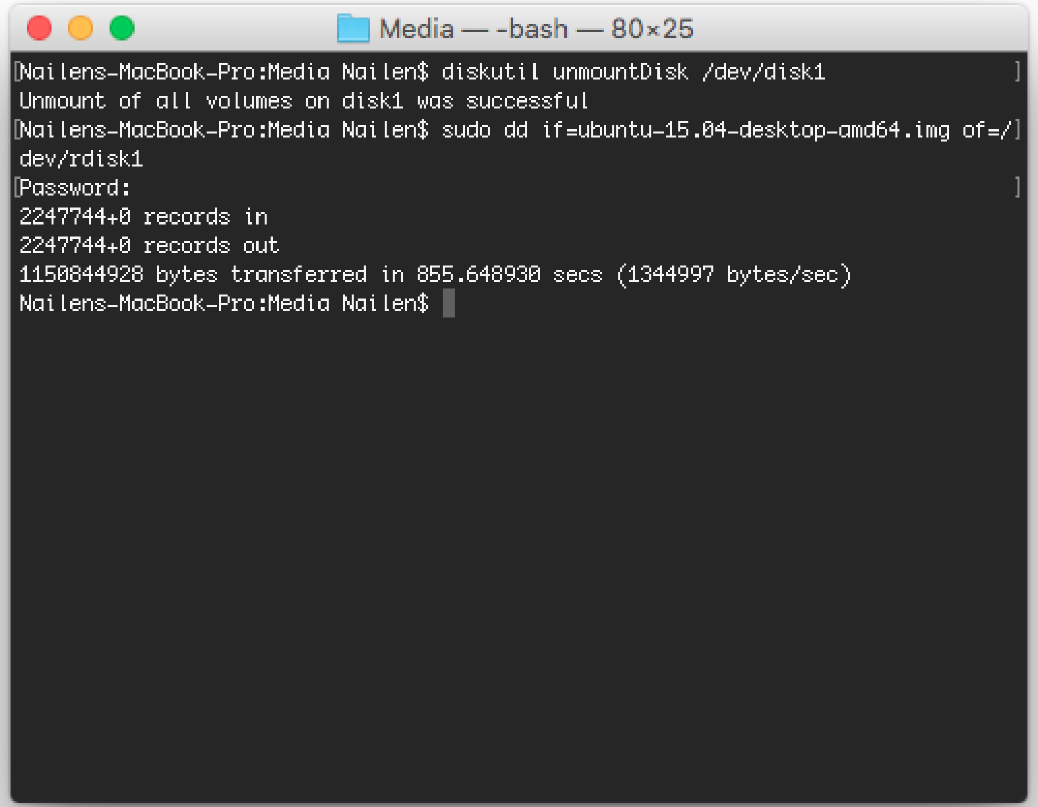
where N is the number at the end of the drive's path, which in my case is $N = 1$. Now, your drive is ready to be made into a bootable live USB for installing Ubuntu. Unfortunately, OS X doesn't play well with the ISO image format that distributions like Ubuntu typically come in, so we have to convert it to an Apple disk image (.dmg) file instead. All this requires is running the following command in terminal:

```
hdiutil convert -format UDRW -o /path/to/image.dmg /path/to/image.iso
```

replacing "/path/to/image" with wherever your file is, e.g. /Users/Nailen/Desktop/Media/ubuntu-15.04-desktop-amd64. This should finish in under a minute, after which we can go ahead and clone this image to the USB drive with:

```
sudo dd if=/path/to/image.dmg of=/dev/rdiskN
```

which will require your password. The reason we use "rdiskN" instead of "diskN" here is that the former basically just writes the data from the start of the disk, whereas the latter has to pass through more layers of software abstraction. This process can take a while, but once it's done you should see a message confirming that the image has been copied, like this:



```
Nailens-MacBook-Pro:Media Nailen$ diskutil unmountDisk /dev/disk1
Unmount of all volumes on disk1 was successful
Nailens-MacBook-Pro:Media Nailen$ sudo dd if=ubuntu-15.04-desktop-amd64.img of=/
dev/rdisk1
Password:
2247744+0 records in
2247744+0 records out
1150844928 bytes transferred in 855.648930 secs (1344997 bytes/sec)
Nailens-MacBook-Pro:Media Nailen$
```

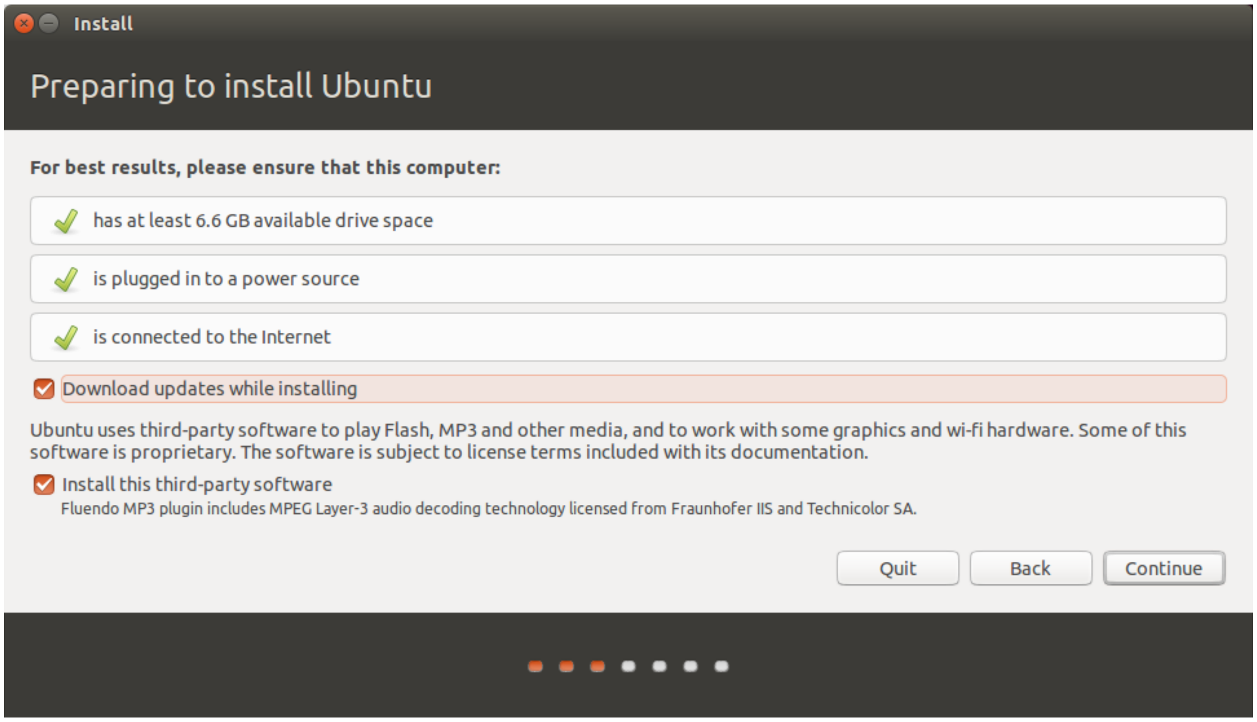
(I used a .img extension instead of .dmg, but it shouldn't matter.) Don't worry if Finder complains about not being able to read the drive, that's normal. If you made it this far, your USB drive should be a bootable instance of Ubuntu, from which you can install the OS on your computer itself.

Part 2: Preparing your Mac

The next step is to create space on your Mac's hard drive for Ubuntu. Go back to Disk Utility, and select your computer's built-in hard drive under the "Internal" category, which will likely have a name along the lines of "APPLE HDD HTS727575A9E362 Media." Make sure you're looking at the drive itself, and not one of its partitions, e.g. "Macintosh HD," and from there, select the "Partition" option in the top bar, which will show a drop down menu to partition the disk. We actually have to create two partitions: a small boot loader that your Mac's firmware will read at startup, and a large one to hold all of your Ubuntu files. Hit the "+" button to create a new partition, and make it 128 MB with the "OS X Extended (Journaled)" format, and a name such as "Ubuntu Boot Loader." Hit "Apply," and then create another partition with a name you'll remember, and a decent amount of size - 50 GB should be enough, but it doesn't hurt to go with more if you can spare it. It doesn't matter what the partition's format is, since we'll be changing that during the install process anyway. Go ahead and hit "Apply," and wait a couple minutes for the process to complete. Your computer might be a bit unresponsive during the partitioning, which is also normal.

Part 3: Installing Ubuntu

Now that your computer has a space for Ubuntu, and you have a flash drive to install it with, it's time to reboot and install. Reboot your computer, and hold down the "Option" key until you see a grey screen with an image of a hard drive named "Macintosh HD," and two orange flash drives named "EFI Boot." Selecting either of the latter two should bring you to a black screen with a few options, including "Try Ubuntu" and "Install Ubuntu." Choose the former, which should already be selected, and your computer will boot into the Ubuntu desktop. There should be an icon which says "Install Ubuntu," which you should double-click to bring up the installer. Proceed through the installation process, selecting "Download updates while installing" and "Install this third-party software" on this screen:


Install

Preparing to install Ubuntu

For best results, please ensure that this computer:

✓ has at least 6.6 GB available drive space

✓ is plugged in to a power source

✓ is connected to the Internet

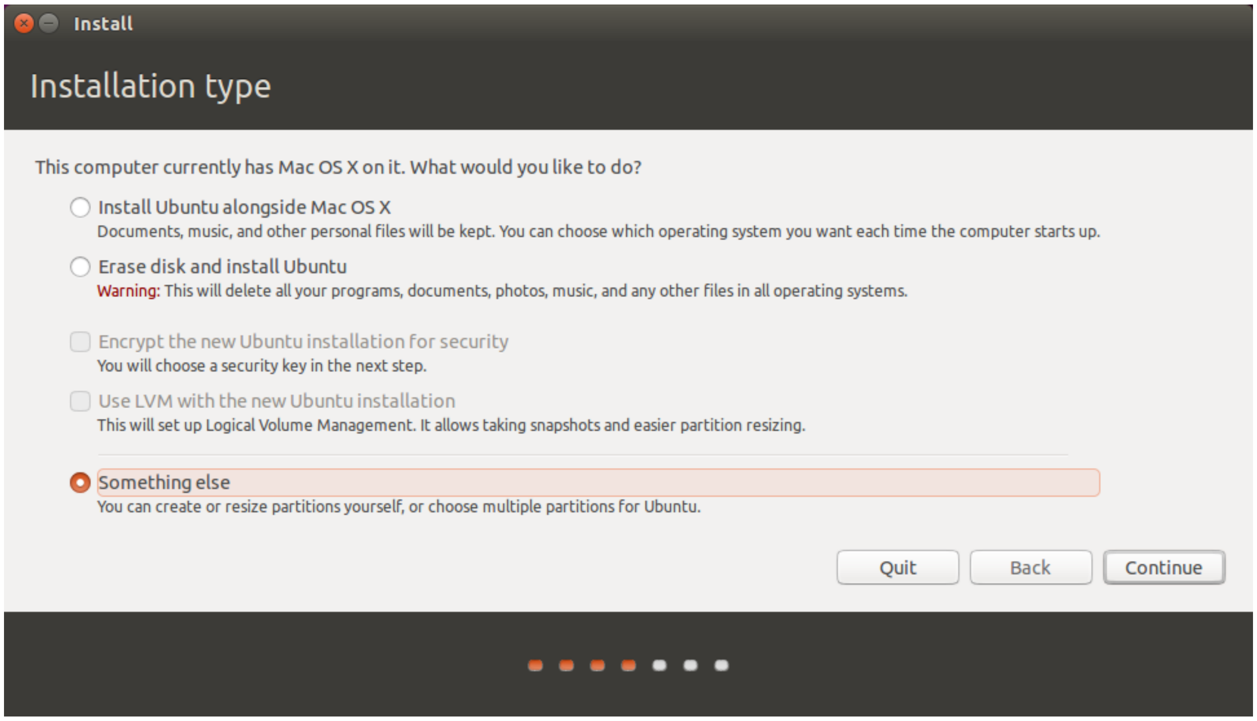
[✓] Download updates while installing

Ubuntu uses third-party software to play Flash, MP3 and other media, and to work with some graphics and wi-fi hardware. Some of this software is proprietary. The software is subject to license terms included with its documentation.

[✓] Install this third-party software

Fluendo MP3 plugin includes MPEG Layer-3 audio decoding technology licensed from Fraunhofer IIS and Technicolor SA.

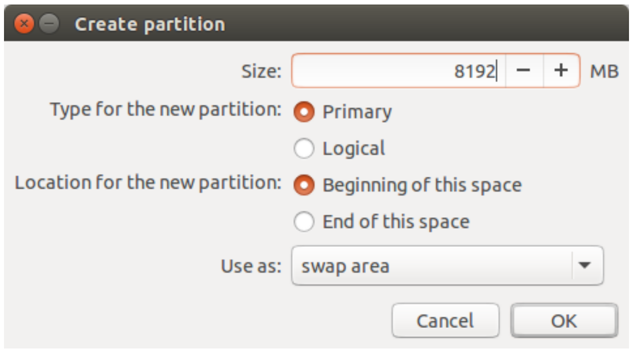
When you get to the "Installation type" step, STOP! This is the point where we configure that partition we made earlier for Ubuntu. Select "Something else," like so, and continue:



This computer currently has Mac OS X on it. What would you like to do?

- ○ Install Ubuntu alongside Mac OS X
  Documents, music, and other personal files will be kept. You can choose which operating system you want each time the computer starts up.
- ○ Erase disk and install Ubuntu
  Warning: This will delete all your programs, documents, photos, music, and any other files in all operating systems.
- ☐ Encrypt the new Ubuntu installation for security
  You will choose a security key in the next step.
- ☐ Use LVM with the new Ubuntu installation
  This will set up Logical Volume Management. It allows taking snapshots and easier partition resizing.
- [✓ Something else]
  You can create or resize partitions yourself, or choose multiple partitions for Ubuntu.

This will bring you to a rudimentary partition editor. Find the partition you created earlier, which should have the same size, be of format HFS+, and be almost entirely empty; click on it, and then click "-" to erase it to free space. If you're still not sure you have the right partition, you can use the pre-installed GParted Partition Editor to finish this section, as it has a similar interface but provides more detail. Now, we have to create at least two partitions: one to use as a swap for your RAM, and one to use for everything else. To do this, make sure the free space is selected, and hit the "+" button. This will bring up a window like this:



Size: 8192 MB

Type for the new partition: [✓ Primary] Logical

Location for the new partition: [✓ Beginning of this space] End of this space

Use as: swap area

Set the partition size to the same as your RAM (remembering that 1 GB = 1024 MB), and the "Use as:" field to "swap area," then hit "OK." Repeat this process, instead letting the size be as large as the remaining free space, and selecting the partition for use as an "ext4 filesystem," with the mount point /. Make a note of the index of this partition, i.e. the N in /dev/sdaN. Those who have used Linux before might also at this point want to create separate partitions for /boot, /home, et cetera, which isolates disk errors and allows parts of the filesystem to be encrypted independently, but this is not necessary. Once you're done, select the ext4 partition as the "Device for boot loader installation," and continue with the rest of the installation. When it finishes, you'll be prompted to reboot or continue testing Ubuntu; you should opt not to reboot since we still have one thing left to do.

Part 4: Making Ubuntu bootable

We're now at the home stretch. Ubuntu has been installed on your computer, but you won't be able to boot into it since there's no EFI boot loader, needed for Mac firmware to recognize the OS as bootable. So, we need to go into the Ubuntu installation and create a boot.efi file that we'll use to boot it. While still inside your live USB instance of Ubuntu, open up a Terminal window, and run the following command to mount the ext4 partition you created at the path /mnt:

```
sudo mount /dev/sdaN /mnt
```

You'll also need to mount several of the top-level directories, which is easiest to do with the command:

```
for i in /dev /dev/pts /proc /sys /run; do sudo mount -B $i /mnt$i; done
```

Now, we can use the `chroot` command to change our root directory to /mnt, so that the bash session in the Terminal is effectively running inside the Ubuntu installation:

```
sudo chroot /mnt
```

This allows us to configure GNU GRUB, the "GRand Unified Bootloader," with:

```
grub-mkconfig -o boot/grub/grub.cfg
```

and then save this to a standalone file with:

```
grub-mkstandalone -o boot.efi -d usr/lib/grub/x86_64-efi -O x86_64-efi --compress=xz boot/grub/grub.cfg
```

You can then exit the `chroot`ed partition with the `exit` command. The boot.efi file is still in the new installation, however, so you'll have to use the `cp` command to copy it to somewhere like ~/Documents/, from which you can move it to somewhere you can access it from OS X, such as another USB drive or Google Drive:

```
cp /mnt/boot.efi ~/Documents/
```

Once you've saved this file, you can reboot your Mac to OS X.

Back in OS X, we want to open up a Terminal window, and `cd` into the boot loader partition we created earlier:

```
cd /Volumes/Ubuntu\ Boot\ Loader/
```

In order to make this recognizable by Apple's firmware, we have to create the mach_kernel and System/Library/CoreServices directories, the latter of which will store our boot.efi file from before, along with a "SystemVersion.plist" file which provides configuration information. The commands to do this are as follows:

`sudo mkdir System mach_kernel`
`cd System`
`sudo mkdir -p Library/CoreServices` (the `-p` option tells `mkdir` to create intermediate directories)
`cd Library/CoreServices`

Now, we copy boot.efi to the CoreServices folder. If you uploaded it to Google Drive like I did, you can then download it from OS X, and then copy it from your Downloads folder with:

```
sudo cp ~/Downloads/boot.efi ./
```

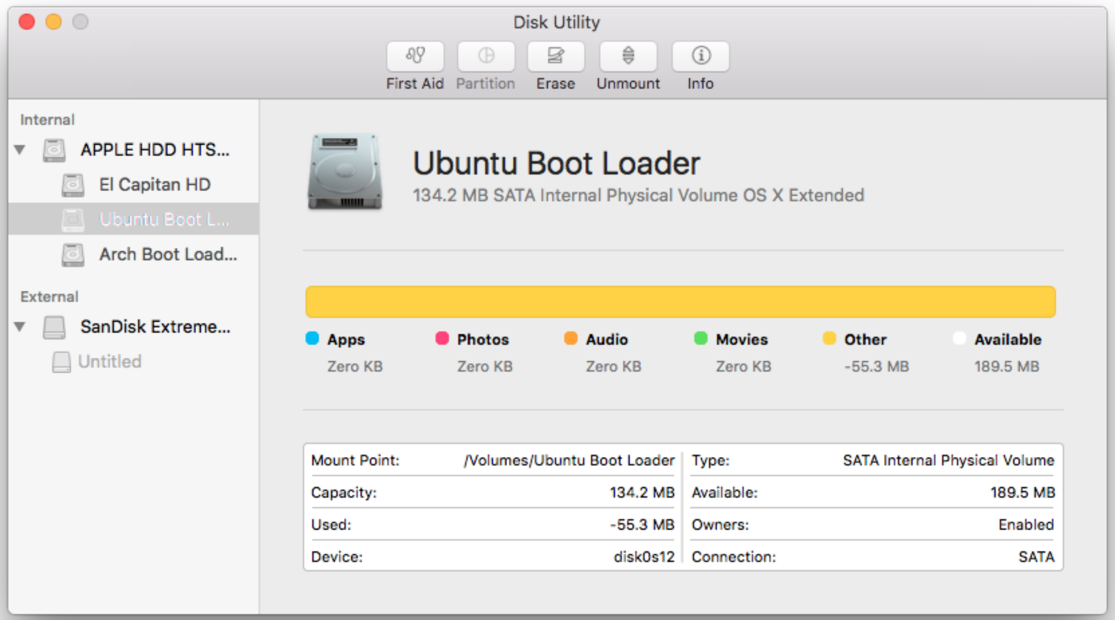
Then, we have to create the SystemVersion.plist file. If you're familiar with a command-line text editor such as vim, you can just run it with `sudo`; otherwise the easiest way to do this is probably to create the file in your Documents folder with TextEdit, and then copy it over via the same method as boot.efi. Either way, it should contain the exact following contents:

```
<xml version="1.0" encoding="utf-8"?>
<plist version="1.0">
<dict>
   <key>ProductBuildVersion</key>
   <string></string>
   <key>ProductName</key>
   <string>Linux</string>
   <key>ProductVersion</key>
   <string>Ubuntu Linux</string>
</dict>
</plist>
```

This defines a dictionary of key-value pairs identifying the Ubuntu OS. The indents can be done with tabs or spaces; they're purely stylistic, as the XML tags will be parseable even without indentation. Once this is in the CoreServices folder, we're almost done! All we have to do is "bless" the partition for booting, and set the boot flag, like so:

```
sudo bless --device /dev/disk0sN --setBoot
```

In this case, N **is the index of the boot loader partition**; because I have a bunch of partitions for my Arch Linux installation, and I accidentally created the boot loader partition after installing Ubuntu, $N = 12$ on my machine. You should check that you have the right number, and that the partition is also on disk0, by opening up Disk Utility again, and looking at the "Device" field of the Ubuntu Boot Loader partition:



Ubuntu Boot Loader

134.2 MB SATA Internal Physical Volume OS X Extended

Apps	Photos	Audio	Movies	Other	Available
Zero KB	Zero KB	Zero KB	Zero KB	-55.3 MB	189.5 MB

Mount Point:	/Volumes/Ubuntu Boot Loader	Type:	SATA Internal Physical Volume
Capacity:	134.2 MB	Available:	189.5 MB
Used:	-55.3 MB	Owners:	Enabled
Device:	disk0s12	Connection:	SATA

The `--setBoot` flag sets Ubuntu partition as the default boot device, so if you'd prefer to keep OS X as your primary OS, just don't include it in the command.

If you've made it this far, your Ubuntu installation should work! Unplug the USB installation drive, and give it a try by rebooting your computer, while holding down the "option" key. In addition to your Macintosh HD, you should see a new image of a hard drive, labeled "EFI Boot." This is Ubuntu, and selecting it will allow you to boot to your new Linux desktop.