"Creating a Linux Multi-Boot DVD"

Lawrence Wellman
Computer Information Technology
Brigham Young University – Idaho
Rexburg, ID, 83440, USA
wel06011@byui.edu

Abstract

Linux live CD's provide a way to run the Linux operating system upon boot, without installing on the hard drive. Contained herein is an introduction into the combining of many Linux live CD's onto a single Multi-Boot DVD.

1 INTRODUCTION

I have had an interest in combining many Linux distributions onto one DVD. There are a few programs out there that will do this, but most only run under Windows and are limited in the distributions that they support. I have endeavored to create a Multi Boot DVD using freely available tools in Linux. The process that is discussed will work for most distributions of Linux, but specific distributions may be different and require more in-depth study into that distribution.

1.1 Benefits of a Live DVD

Having an operating system that is capable of running in memory without having the requirement of installing on the hard drive can have many benefits. It gives the ability to test Linux to those who have never experienced it or those who are trying a new distribution. It also allows one to test for compatibility issues with their specific system without having to install. Many users use Live CDs to do system repair and restoration. It can be used for forensics allowing you to run an operating system and tools without actually touching the hard drive. Another popular use is for system and network security testing, or penetration testing.

There are many Linux distributions available which have different strengths and weakness in how they run, what they support, and what tools they provide. Being able to carry multiple versions on one DVD allows you to have multiple distributions so you can have what you need on one DVD without having to carry a case of CD's with each individual distribution.

1.2 Requirements and Suggestions

Creating a Linux Multi-Boot DVD with out the use of an automated tool is not simple and will take a basic understanding of Linux and its commands. I will describe what most of the commands are doing but for further adaptation to your system or your chosen distributions you may have to change the commands to suit your situation. Some knowledge of Linux boot process and distributions will make things run smother and make it easier to find required files.

You will need approximately 10 GB of free space, ≈5 GB are for the Linux distributions you want on your live DVD and the other 5 GB are for the finished product.

I suggest installing a virtual machine (VMware or Virtual Box); this allows you test your DVD and make any changes that need to be made without wasting a DVD.

2 GETTING STARTED

Before we begin, we will want to get our computer ready to create our Live DVD.

2.1 Select your Live CD Distributions

There are many live distributions; www.frozentech.com/content/livecd.php contains a fairly extensive list of live CD distributions. It gives basic information about what the distribution is meant for and a link to the site where you can get more information on the distribution and download the distribution of your choice. For this paper I used Backtrack, Ophcrack, nUbuntu, Helix, Insert, and Damn Small Linux.

2.2 Create Folder Structure

You will now need to create some folders where we will store our files and use them to organize our DVD. Start by creating a folder named 'dvdroot' then as subfolders you will need a folder 'isolinux.' We will add a folder for each distribution as we add them to the DVD. You will also want to create folders to use as mount points; these can be named whatever you want. I chose to use the name of the distribution, just to keep track of which distribution I was working with.

3 CREATING THE DVD

We will now create the DVD[1] by pulling the files from each individual distribution and then adding them to our folder structure.

3.1 Mount the iso's

Start by mounting the DVD. This can be done with the following command:

mount –o loop [src] [mountpoint]
Example: mount –o loop /bt2final.iso /backtrack/

(Note: As mentioned earlier, it is best to do the distributions individually but if decided to do more than one at a time you may need to manually increase the number of loopbacks on your computer to allow you to mount more iso's; generally if you are mounting more than 8 iso's at a time.)

3.2 Pilfering files and bending them to our will.

After you mount the iso you will need to open the iso. There are several files that we will need and they are not always named the same. There will be two categories of files that we need:

1. Operating system files: There is a little more play in the name of these files. Generally there will only be two folders so you can deduce which one this is. Commonly it is the name of the distribution but not always.
2. Live boot files: these will generally be in a folder called either boot or isolinux.

3.3 Operating System files

The operating system files will be copied over without making in changes if possible. Most likely you will have a few distributions that will have the same folder name, thus requiring you to change the folder's name.

3.4 Boot Files

There are several files we will need from the boot folder. Some files we will take unaltered, others we will need to make changes to or take information out of. We will place these files in our isolinux folder. We need one each these file may be taken from any distribution and are taken unaltered:

- isolinux.bin - boot loader for no-emulation
- isolinux.boot - required for isolinux

Files that require information or altering:

- boot.msg - contains a message displayed on boot
- isolinux.cfg - configuration file
- vmlinuz (from each distro) - is the name of the Linux kernel executable
- .gz file (from each distro) - contains info of where to find linux files

3.4.1 vmlinuz

The vmlinux file is a compressed bootable Linux kernel. We need the vmlinuz from each distribution. We will need to rename these so there are no naming conflicts. The easiest way is to give each an extension; make sure that it is no more than 3 characters. For example vmlinux.bt – for Backtrack or vmlinuz.nu – for nUbuntu

3.4.2 .gz file

There will be a file with the .gz extension. Most common is initrd.gz; we will need this file. It will mostly likely need to be renamed to prevent naming conflicts. The name will need to be 8 or less characters. (ex. initrdbt.gz – Backtrack) If you copied over the operating system files without changing the name then there are no other changes that need to be made to this file.

If, however, you had to change the name, we will need to make some changes to this file. You will need to open the inird.gz file which can be done using gunzip (ex. gunzip initrd.gz). You will then need to mount the initrd file (ex. mount –o loop initrd /mount_initrd/). After you mount the file you will need to search for a file that declares the dir to be the name of the folder that you changed, most likely named linuxrc. Once you find the file, you will need the name to change to point to the new file name.(ex. KNOPPIX_DIR='KNOPIX' change to KNOPPIX_DIR='helix'). You will then need to recreate the .gz file by un-mounting the file (umount /testmount) and then re-zipping it (gzip initrd).

3.4.3 isolinux.cfg

You only need one isolinux.cfg in your isolinux folder. This file contains the boot options used for booting your different distributions. You can create your own file, but it tends to work best if you copy over 1 file and then copy the Label information from each individual isolinux.cfg. I have added an example isolinux.cfg in Appendix B. [2]Some basic commands when working with this file:

- # - Comment
- DEFAULT – Specifies what will boot after boot timeout or if user just hits ENTER
- TIMEOUT {x} – sets timeout in 1/10 sec

- DISPLAY {filename} – Displays text contained with in file (boot.msg)
- LABEL {label name} – Explains boot options for {label name} (aka if the user types {label name} it will boot with options declared after declaration)
  - KERNEL – kernel to use when {label name} is typed
  - APPEND – declares additional parameters to the kernel declaration

3.4.4 boot.msg

The file you will create will be displayed on boot and will give the user their options; it is a text message. Some design can be done here but is not covered in the paper. An example of a simple text message boot.msg:

4 CREATING THE ISO

Now that we have everything in the folders we need, it is time to make the iso. This is done with the following command [1]:

sudo mkisofs -r -ldots -J -V "Multiboot DVD" -b isolinux/isolinux.bin -c isolinux/boot.cat -no-emul-boot -boot -load-size 4 -boot-info-table -x lost+found -o /multiboot.iso .

An explanation of this command, for those who want to know what each switch does, is included in Appendix A.

5 TESTING

Now that we have created the iso we will need to test it. This can be done by mounting the iso in your choice of virtual machine then starting the virtual machine and see if it starts up and if you access all your different distributions. The benefit of using a virtual machine is that you can test it after you add each individual distribution which will make troubleshooting much easier.

6 TROUBLESHOOTING

Not all distributions or live CD's set up the same. You may run into some that you will require you to set your DVD up a little differently than I have described. If having trouble finding which files are required in the boot process, boot the single live CD into your virtual machine and see what files it uses when booting.

Some distros (Mandriva) codes the name of the cd into initrd which you will need to change with a hex editor. Others like Fedora declare the cdlabel in the isolinux.cfg so this will need to be changed to the label of your DVD. If you used the mkisofs command it should be "Multiboot DVD".

If all else fails, or maybe even before that, consult Google and see if someone else has tried adding your desired distribution to a Multi Boot DVD. I also found many distributions have a bulletin board where there are people willing to help with this kind of project.

7 REFERENCES

[1] "Creating a Multi-boot DVD" pcquest.ciol.com/content/enterprise/2005/105070101.asp
[2] "Isolinux HowTo for newbies" members.chello.at/bobby100/ILpart1.htm
[3] "Making a Multiple-Boot CD" http://linuxgazette.net/issue85/sipos.html
[4] "Building Your Own Live Cd" http://www.linuxjournal.com/article/7246

8 Appendix A

The command which makes the iso may be a bit complicated so I have listed what each of the switches do.

- sudo Allows us to execute the command as root
- mkisofs Create a hybrid ISO9660/JOLIET/HFS filesystem
- -r Uses naming conventions for compatibility and makes all files publicly readable
- -ldots Allow ISO9960 filenames to begin with a period.
- -J Uses Joliet naming records, for Windows compatibility
- -V "Multiboot DVD" Provides a volume ID, this is the disk name that shows up in Window's Explorer
- -b isolinux/isolinux.bin Specifies the path and filename of the boot image to be used when making a bootable cd.
- -c isolinux/boot.cat Specifies the path and filename of the boot catalog to be used when making a bootable cd.

- -no-emul-boot Specifies that it is not a image of a floppy
- -boot-load-size 4 Specifies the number of “virtual” sectors to load in no-emulation mode.
- -boot-info-table Specifies that a 56-byte table with info of the CD-ROM layout will be patched in at offset 8 in the boot file
- -x lost+found Exclude path from being written to CDROM
- -o /multiboot.iso Names the new .iso image file
- . Specifies the path to the source for the image

9 Appendix B

```
PROMPT 1
TIMEOUT 400
DEFAULT bt

display boot.msg

LABEL bt
KERNEL /isolinux/vmlinuz.bt
APPEND vga=0x317 max_loop=255 initrd=/isolinux/initrdbt.gz init=linuxrc load_ramdisk=1 prompt_ramdisk=0
ramdisk_size=4444 root=/dev/ram0 rw

LABEL opv
KERNEL /isolinux/vmlinuz.opv
APPEND initrd=/isolinux/initrdov.gz ramdisk_size=6666 root=/dev/ram0 rw autoexec=xconf;startx changes=/
slax/

LABEL opx
KERNEL /isolinux/vmlinuz.opx
APPEND initrd=/isolinux/initrdox.gz ramdisk_size=6666 root=/dev/ram0 rw autoexec=xconf;startx changes=/
slax/

LABEL nubuntu
KERNEL /isolinux/vmlinuz.nu
APPEND  file=/cdrom/preseed/ubuntu.seed boot=casper initrd=/casper/initrdnu.gz ramdisk_size=1048576
root=/dev/ram rw quiet splash --
```