fomori blog

[adventures in nerdism]

Create a GNU/Linux multiboot USB-Stick (Live USB) with Syslinux chainloading

Posted on **January 25, 2014** by **6arms1leg**

Since USB-Sticks, that are fast and have a high capacity, are finally affordable, I decided to buy a new one. I usually install a GNU/Linux live CD (more precisely live USB) distribution on my USB-Sticks: either SystemRescueCd or Kali Linux (former Backtrack). The left over space is used for the classical purpose of an USB-Stick – data exchange. Todays USB-Sticks have enough capacity to easily fit several GNU/Linux live distributions on them, while still leaving enough space for other data. So my plan was to create a multiboot USB-Stick, that would boot my favourite GNU/Linux live distributions mentioned above. Unfortunately, searching the internet for implementing this did not give me any satisfactory results. There are a ton of guides that explain how to create an USB-Stick that boots GNU/Linux, but there are almost no multiboot solutions. The few howto's about multiboot USB-Sticks are either about booting *.iso files (which only works with some GNU/Linux distributions) with GRUB 2 (which is designed for static boot setups anyway) or require further customized modifications of the GNU/Linux live distributions. I wanted a simpler solution that – once created – allows for easy updating of the installed GNU/Linux live distributions.

This guide will explain how to create a multiboot USB-Stick that can boot several GNU/Linux live dirstibutions via Syslinux chainloading. It will have several partitions (one for each OS and one for the main Syslinux bootloader) and a separate data partition, that can be formated independently in any way you like, so that your data is seperated from the operation system data. This guide installs SystemRescueCD and Kali Linux on your multiboot USB-Stick, but any other GNU/Linux live distribution should work as well. Adding more than two OS should also be no problem.

Overview

The idea is to create four partitions: The first is the data partition, because MS Windows only shows the first partition of an USB-Stick, so under MS Windows your multiboot USB-Stick will appear as an ordinary USB-Stick with the size of your data partition. The second partition is a small boot partition with the “boot” flag and the “main” Syslinux bootloader on it, that will chainload the other bootloaders of your GNU/Linux live distributions. The third and fourth partition are used for the live distributions and their own Syslinux bootloaders installed to the VBR. Also, the Syslinux boot code will be copied to the MBR of the USB-Stick.

When you boot the USB-Stick, the boot code of the MBR will be executed and looks for the partition marked bootable. This partition will then be booted. Since the boot partition only has the “main” Syslinux bootloader on it with the chainload entries, that point to the bootlaoders of each partition, you will be presented with the “main” boot menu. From within this boot menu you can choose the OS you want to boot.

Hardware requirements

USB Stick

All you need is an USB-Stick that has enough space to install the GNU/LInux live distributions on it and leaves some extra room for your data.

Computer

Most GNU/Linux live distributions are designed to boot x86 or x64 devices, so the computers you want to boot with your multiboot USB-Stick probably should have one of those architectures. Additionally, the computers must be capable of booting USB devices and the first boot device must be set to “USB”.

Get GNU/Linux live disributions

First, you have to download your desired GNU/Linux live distributions and check their integrity.

SystemRescueCd

Go to http://www.sysresccd.org/Download and download the latest stable release, e.g.:

```
$ wget http://downloads.sourceforge.net/project/systemrescuecd/sysresccd-x86/3.8.
```

After the download has finished, compare the sha256sums. Type

```
$ sha256sum systemrescuecd-x86-3.8.1.iso
```

and compare the output to the sha256sum for your download on http://www.sysresccd.org/Download. The sums should be identical, otherwise your downloaded file is corrupt.

Kali Linux

This procedure is quite similar for Kali Linux. Go to http://www.kali.org/downloads/ and download the latest stable release, e.g.:

```
$ wget http://cdimage.kali.org/kali-latest/i386/kali-linux-1.0.6-i386.iso
```

After the download has finished, compare the sha1sums. Type

```
$ sha1sum kali-linux-1.0.6-i386.iso
```

and compare the output to the sha1sum for your download on http://www.kali.org/downloads.

Calculate partition sizes in percent

This step is needed for optimal alignment during formatting with parted in the next step. If “relative” values in percent are given for “start” and “end” during formatting, parted will align partitions automatically when started with the “–align optimal” switch (see next section).

Check the needed size for each OS, so you know how much space they need for installation on your USB-Stick. In this example, SystemRescueCD needs about 500 MiB and Kali Linux 3200 MiB, so I chose 1000 MiB for SystemRescueCD and 4000 MiB for Kali Linux, respectively, to leave some room if future updates of the distributions require more space. For the boot partition 100 MiB should be more than enough. The size of my USB-Stick is 61057 MiB. Adjust these values to your setup.

Now, calculate the partition end points “x” in percent. The easiest way is to calculate the occupied space of the following partitions (in percent) and substract it from the total space available. The following will give you the end points of your partitions. The start point of each partition is the end point of the preceding partition (for the first partition 0 %). The mathematics is not really worth mentioning, but anyway:

- Data partition (following partitions: Boot, SystemRescueCd, Kali Linux)

$$61057\ \mathrm{MiB} * x = 100\ \mathrm{MiB} + 1000\ \mathrm{MiB} + 4000\ \mathrm{MiB} => x = 5100\ \mathrm{MiB} / 61057\ \mathrm{MiB}$$
$$1 - x = 0{,}916 \equiv 91{,}6\ \%$$

- Boot partition (following partitions: SystemRescueCd, Kali Linux)

$$61057\ \mathrm{MiB} * x = 1000\ \mathrm{MiB} + 4000\ \mathrm{MiB} => x = 5000\ \mathrm{MiB} / 61057\ \mathrm{MiB}$$

$1 - x = 0{,}918 \equiv 91.8\ \%$

- SystemRescueCd partition (following partitions: Kali Linux)

$61057\ MiB * x = 4000\ MiB => x = 4000\ MiB / 61057\ MiB$
$1 - x = 0{,}934 \equiv 93{,}4\ \%$

- Kali Linux partition (following partitions: none)

$61057\ MiB * x = 0\ MiB => x = 0\ MiB$
$1 - x = 1 \equiv 100\ \%$

Partition the USB-Stick

Warning: This step will partition your USB-Stick. All data on it will be lost! Also proceed with extra caution when typing the device name! Entering the wrong device name may result in data loss on other storage devices.

First, find out which device your USB-Stick is. If you are not sure, type

```
$ dmesg
```

after plugging in the USB-Stick.

To partition the USB-Stick, start parted (replace “/dev/sdX” with your actual device!):

```
$ sudo parted --align optimal /dev/sdX
```

You should now see the parted prompt, “(parted)”. To see your current partition table, type:

```
(parted) print
```

Set the partition layout to “MBR” and set parted’s standard unit to percent (“%”):

```
(parted) mklabel msdos # Set MBR layout.
(parted) unit % # Set size unit of parted to percent, needed for correct alignmen
```

Now, create the four partitions. By choosing the start point of each following partition a little smaller than the preceding one, you make sure parted leaves no gaps between the partitions. Do not worry, parted will notice that the partitions overlap and suggests the closest possible match. Just confirm these actions each time by typing “yes”.

```
(parted) mkpart primary fat32 0 91.6 # data partition
(parted) mkpart primary ext2 91.5 91.8 # boot partition
(parted) mkpart primary fat32 91.7 93.4 # SystemRescueCD partition
```

```
(parted) mkpart primary ext2 93.3 100 # Kali Linux partition
```

The boot partition must be marked as bootable. To set the bootable flag, do:

```
(parted) set 2 boot on # Mark second partition (boot partition) as bootable.
```

Finally, check alignment and size of each partition:

```
(parted) align-check optimal 1 # Check data partition alignment.
(parted) align-check optimal 2 # Check boot partition alignment.
(parted) align-check optimal 3 # Check SystemRescueCD partition alignment.
(parted) align-check optimal 4 # Check Kali Linux partition alignment.
(parted) unit mib # Set unit back to MiB for better readability.
(parted) print # Show current partition table to check partition sizes.
```

If everything is as expected, type

```
(parted) quit
```

to exit parted.

Format USB-Stick

Warning: This step will format your USB-Stick. All data on it will be lost! Also proceed with extra caution when typing the partition name! Entering the wrong partition name may result in data loss on other storage devices.

For the data partition FAT32 is probably the best choice, as it is readable by virtually all operating systems. Feel free to pick a different file system, if you want. For the boot and Kali Linux partition, ext2 should be used. Ext2 has no journaling and thus reduces write cicles on your USB-Stick (in contrast to ext3/4). This guide uses the Extlinux sub-programm of the Syslinux package, so the filesystem must be ext2/3/4. However, it is also possible to choose FAT32 for the boot partition (but that is not covered in this guide). The Kali Linux file system needs to be ext2, because Kali Linux makes use of symlinks, which are not supported with FAT32. I chose FAT32 for the SystemRescueCd file system, simply because it is used in the howto's on their website (and their install script on the *.iso file). It would be perfectly possible to use an ext2 file system (but that also is not covered in this guide). If you do not stick to the file system setup in this guide, remember to adjust all following sections.

To format the four partitions you just created, issue (replace "/dev/sdX1"):

```
$ sudo mkfs.vfat -F 32 -n DATA /dev/sdX1 # Create FAT32 data partition. Use upper
$ sudo mke2fs -t ext2 -L boot /dev/sdX2 # Create ext2 boot partition.
$ sudo mkfs.vfat -F 32 -n SYSRESCCD /dev/sdX3 # Create FAT32 SystemRescueCD parti
$ sudo mke2fs -t ext2 -L kalilinux /dev/sdX4 # Create ext2 Kali Linux partition.
```

Install Syslinux with chainloading

This step will install Syslinux to the MBR and to the boot partition and configure it to chainload the live distributions.

One important general information about Syslinux: The Syslinux COM32 modules are not interchangable between different Syslinux versions! Always make sure to use the correct Syslinux version matching your Syslinux COM32 files and vice versa — or it may not work! (This is why it is necessary to download an old version of Syslinux to install Kali Linux later in this guide.)

First, make sure you have the “syslinux” and “mtools” package installed. If not, install it via your GNU/Linux distributions package manager, e.g. for Arch Linux:

```
$ sudo pacman -S syslinux # Install "syslinux" package.
$ sudo pacman -S --asdeps mtools # Install "mtools" package, needed to install Sy
```

Or Debian:

```
$ sudo aptitude install syslinux # Install "syslinux" package.
$ sudo aptitude install mtools # Install "mtools" package, needed to install Sysl
```

It is also possible to download the “syslinux” binary package manually and install Syslinux to the MBR and partitions (see commands below) without installing it on your operating system. This will be needed to install Kali Linux, anyway (later in this guide).

Now, mount the boot partition, create the Syslinux directory (which will hold some of the important Syslinux data) and copy all COM32 modules to it (actually, only “vesamenu.c32″ and “chain.c32″ are needed). Unfortunately, the location of the COM32 modules and the Syslinx MBR binary file may differ depending on your GNU/Linux distribution. The “cp” command below uses the location on Arch Linux. The COM32 modules are also distributed with the syslinux binary package. (replace “/devsdX2″, “/mountpoint” “/usr/lib/syslinux/bios/*.c32″ and “/usr/lib/syslinux/bios/mbr.bin” with your device’s partition, mount point, COM32 modules location and Syslinux MBR binary file location, respectively.)

```
$ sudo mount /dev/sdX2 /mountpoint
$ sudo mkdir /mountpoint/syslinux
$ sudo cp -r /usr/lib/syslinux/bios/*.c32 /mountpoint/syslinux/ # The source loca
```

Install Syslinux (using the “extlinux” command because of the ext2 file system) to the partition’s VBR and “syslinux” directory:

```
$ sudo extlinux --install /mountpoint/syslinux
```

and install the MBR to the device:

```
$ sudo dd bs=440 count=1 conv=notrunc if=/usr/lib/syslinux/bios/mbr.bin of=/dev/s
```

Finally, create the Syslinux configuration file and copy the content below:

```
$ sudo nano /media/ext_device_2/syslinux/syslinux.cfg # Create Syslinux configura
```

add (feel free to change the values here to change the boot menu layout):

```
UI vesamenu.c32
DEFAULT sysresccd
PROMPT 0
MENU TITLE Multiboot USB-Stick - Boot Menu
TIMEOUT 50

MENU WIDTH 80
MENU MARGIN 10
MENU PASSWORDMARGIN 3
MENU ROWS 12
MENU TABMSGROW 18
MENU CMDLINEROW 18
MENU ENDROW -1
MENU PASSWORDROW 11
MENU TIMEOUTROW 20
MENU HELPMSGROW 22
MENU HELPMSGENDROW -1
MENU HIDDENROW -2
MENU HSHIFT 0
MENU VSHIFT 0

#MENU BACKGROUND 37;40 #00000000 #00000000 none

# Refer to http://www.syslinux.org/wiki/index.php/Comboot/menu.c32

menu color screen    37;40      #80ffffff #00000000 std
menu color border    30;44      #40000000 #00000000 std
menu color title    1;36;44    #c00090f0 #00000000 std
menu color unsel    37;44      #90ffffff #00000000 std
menu color hotkey    1;37;44    #ffffffff #00000000 std
menu color sel        7;37;40    #e0000000 #20ff8000 all
menu color hotsel    1;7;37;40  #e0400000 #20ff8000 all
menu color disabled    1;30;44    #60cccccc #00000000 std
menu color scrollbar    30;44      #40000000 #00000000 std
menu color tabmsg    31;40      #90ffff00 #00000000 std
menu color cmdmark    1;36;40    #c000ffff #00000000 std
menu color cmdline    37;40      #c0ffffff #00000000 std
menu color pwdborder    30;47      #80ffffff #20ffffff std
menu color pwdheader    31;47      #80ff8080 #20ffffff std
menu color pwdentry    30;47      #80ffffff #20ffffff std
menu color timeout_msg    37;40      #80ffffff #00000000 std
menu color timeout    1;37;40    #c0ffffff #00000000 std
menu color help        37;40      #c0ffffff #00000000 std
menu color msg07    37;40      #90ffffff #00000000 std

LABEL sysresccd
MENU LABEL SystemRescueCd
COM32 chain.c32
APPEND boot 3

LABEL kali
MENU LABEL Kali Linux
COM32 chain.c32
APPEND boot 4
```

```

LABEL reboot
MENU LABEL Reboot
COM32 reboot.c32

LABEL poweroff
MENU LABEL Power Off
COM32 poweroff.c32
```

Unmount the partition:

```
$ sudo umount /mountpoint
```

Install GNU/Linux live distributions

All that is left to do now is to install the GNU/Linux live distributions.

SystemRescueCd

Mount the previously downloaded *.iso file and the SystemRescueCd partition (Adjust the paths of "/mountpoint1″, "/mountpoint2″, "systemrescuecd.iso" and "/dev/sdX3″.):

```
$ sudo mount -o loop -o exec systemrescuecd.iso /mountpoint1
$ sudo mount /dev/sdX3 /mountpoint2
```

Copy over the SystemRescueCd files and rename Isolinux to Syslinux (Again, adjust the paths.):

```
$ sudo cp -v -r --remove-destination /mountpoint1/* /mountpoint2/
$ sudo mv /mountpoint2/isolinux/isolinux.cfg /mountpoint2/isolinux/syslinux.cfg
$ sudo mv /mountpoint2/isolinux /mountpoint2/syslinux
```

Replace all occurrences of "isolinux" with "syslinux" and set "scandelay" to "5″ (to leave some time for the USB-Stick to settle during hardware detection.):

```
$ sudo sed -i -e 's!/isolinux/!/syslinux/!g' /mountpoint2/boot/grub/grub*.cfg
$ sudo sed -i -e 's!scandelay=.!scandelay=5!g' /mountpoint2/syslinux/syslinux.cfg
```

Unmount the SystemRescueCd partition (Adjust the mountpoint):

```
$ sudo umount /mountpoint2
```

Now, install Syslinux using the binary from the SystemRescueCd. As mentioned before, because of compability reasons with Syslinux' COM32 modules, it is neccessary to have exactly the same Syslinux version that matches the COM32 files on the partition.

```
$ sudo /mountpoint1/usb_inst/syslinux --install --directory syslinux /dev/sdX3
```

Unmount the *.iso file.

```
$ sudo umount /mountpoint1
```

Kali Linux

Installing Kali Linux is quite similar to installing SystemRescueCd.

Mount the previously downloaded *.iso file and the Kali Linux partition (Adjust the paths of "/mountpoint1", "/mountpoint2", "kali-linux.iso" and "/dev/sdX4".):

```
$ sudo mount -o loop -o exec kali-linux.iso /mountpoint1
$ sudo mount /dev/sdX4 /mountpoint2
```

Copy over the Kali Linux files and rename Isolinux to Syslinux (Again, adjust the paths.):

```
$ sudo cp -v -r --remove-destination /mountpoint1/* /mountpoint2/
$ sudo mv /mountpoint2/isolinux/isolinux.cfg /mountpoint2/isolinux/syslinux.cfg
$ sudo mv /mountpoint2/isolinux /mountpoint2/syslinux
```

As mentioned above, because of compability reasons with Syslinux' COM32 modules ("*.c32"), it is neccessary to have exactly the same Syslinux version that matches the COM32 modules on the partition. The Kali Linux *.iso file does not come with a Syslinux binary, so you have to find out which version of Syslinux the installed COM32 modules belong to. Type

```
$ strings /mountpoint2/syslinux/isolinux.bin | grep -i isolinux
```

and the output should give you the version and build date.

Now, go to https://www.kernel.org/pub/linux/utils/boot/syslinux/ and download the matching Syslinux version from the archive, e.g. Syslinux 4.05:

```
$ wget https://www.kernel.org/pub/linux/utils/boot/syslinux/syslinux-4.05.tar.bz2
```

After the download has finished, compare the sha256sums. Type

```
$ sha256sum syslinux-4.05.tar.bz2
```

and compare the output to the sha256sum for your download on https://www.kernel.org/pub/linux/utils/boot/syslinux/sha256sums.asc.

Extract the compressed archive and install Syslinux (with the "extlinux" command because of the ext2 file system) using the binary from the extracted archive (adjust the paths):

```
$ tar -xjf syslinux-4.05.tar.bz2
$ sudo syslinux-4.05/extlinux/extlinux --install /mountpoint2/syslinux
```

Finally, unmount the *.iso file and the USB-Stick partition:

```
$ sudo umount /mountpoint2
$ sudo umount /mountpoint1
```

Done!

This entry was posted in **Hack**, **Hardware**, **Linux**, **Mod**, **Software** and tagged **chainloading**, **GNU/Linux**, **Kali Linux**, **Live CD**, **Live Linux Distribution**, **Live USB**, **Multiboot**, **Syslinux**, **SystemRescueCd**, **USB Stick** by **6arms1leg**. Bookmark the **permalink [http://fomori.org/blog/?p=747]** .

2 THOUGHTS ON “CREATE A GNU/LINUX MULTIBOOT USB-STICK (LIVE USB) WITH SYSLINUX CHAINLOADING”

user_from_reddit
on **February 18, 2014 at 8:43 pm** said:

Hey man i see you prepared really nice material for setting up such thing. But without partitioning and using little tweaks in the actual CMDLINE of kernel, you can boot such systems.

for example for ubuntu-and-derived distros you put
live-media-path etc.
and similar thing for debian-derived things. kali is a debian-based and all debian based distros (except ubuntu) can be booted in folder and or with iso image.
only thing is you need to extract kernel and initrd from iso seperately.

example folder structure of usb
/ -root
|—- debian7/
| |—- debian7-amd64.iso
| |—- vmlinuz
| |—- gtk-initrd.gz
|—- ubuntu1310/
| |—- vmlinuz
| |—- initrd…
| |—- casper/
| |—- filesystem.squashfs
|—- slax64

```
| |—- boot/
| | |—- vmlinuz
| | |—- initrd.gz
| |—- modules/
| | |—- 1207-nano.sb
| |—- rootcopy/
| |—- 01-core.sb
| |—- 02-xorg.sb
| |—- 03-kde.sb
|—- ttylinux/
| |—- vmlinuz
| |—- initrd.gz
| |—- packages/
| | |—- somepackage.tar.gz
|—- archlinux/
| |—- boot/
| | |—- initrd
| |—- rootfs.fs
| |—- vmlinuz
|—- syslinux/
| |—- syslinux.cfg
| |—- vesamenu.c32
| |—- ldlinux.sys
| |—- ldlinux.c32
| |—- libcom32.c32
| |—- mbr.bin
|—- memtest
|—- cpuinfo
```

David Marceau
on **February 19, 2014 at 12:02 am** said:

Great job! Hats off! One of the most interesting and useful articles I have seen in quite a while. Cheers. How about writing another article like this one but for arm developer boards while it’s still fresh in your mind? You should put ads on your blog because your going to get lots of traffic.