

Timo's Rescue CD Set

Build Your Bootable Rescue CD

Howto make 2.88MB el torito floppy disk images

Preface:

I wrote that little recipe because i did not find any useful information on that title on the net. So maybe that will be useful for someone else.

[Do it with lilo](#)

[Do it with syslinux](#)

[Do it with grub](#)

[Or do it with isolinux](#)

[Multi boot images](#)

1. Use lilo as the bootloader

First make a 2.88MB zeroed image file.

```
dd if=/dev/zero of=/tmp/boot288.img bs=512 count=5760
```

Then set up the loop device.

```
losetup /dev/loop0 /tmp/boot288.img
```

Then format it with a filesystem.

```
mkfs.minix /dev/loop0
```

Mount it at /mnt.

```
mount /dev/loop0 /mnt
```

Assemble your boot floppy at /mnt.

```
-----<snip>-----
```

```
# lilo.conf
```

```
boot    =/dev/loop0
```

```
disk    =/dev/loop0
```

```
bios    =0x00                # 2.88MB disk geometry
```

```
sectors =36
```

```
heads   =2
```

```
cylinders =80
```

```
install =/mnt/boot/boot.b
```

```
map      =/mnt/boot/map
```

```
backup   =/dev/null
```

```
default  =RescueSystem
```

```

prompt
delay      =50
timeout    =50
image      =/mnt/bzImage          # your kernel image
label      =RescueSystem
initrd     =/mnt/initrdfs.gz      # initial ramdisk
root       =/dev/ram0
append     ="ramdisk_size=40960" # initial ramdisk size

```

-----<snip>-----

Now execute lilo.

Last but not least copy the `initrdfs.gz` to `/mnt` and `umount /mnt`.

Detach the loop device: `losetup -d /dev/loop0`.

Voila, that's it.

I got that out of the scripts from tomsrootboot disk - <http://www.toms.net/rb>.

1.2 Use syslinux as the bootloader

Syslinux will maybe work on systems where the lilo bootloader does not work.

That's because there are some buggy bioses out in the universe which does not implement the full eltorito standard. And BTW syslinux is a lot easier to set up then lilo.

First make a 2.88MB zeroed image file.

```
dd if=/dev/zero of=/tmp/boot288.img bs=512 count=5760
```

Then set up the loop device.

```
losetup /dev/loop0 /tmp/boot288.img
```

Then format it with the fat filesystem because syslinux only operate from dos formatted disks.

```
mkfs.msdos /dev/loop0
```

Mount it at `/mnt`.

```
mount /dev/loop0 /mnt
```

-----<snip>-----

```
# syslinux.cfg
```

```
DEFAULT    RescueSystem
```

```
PROMPT     1
```

```
LABEL      RescueSystem
```

```
KERNEL     bzImage
```

```
APPEND     initrd=initrdfs.gz ramdisk_size=40960 root=/dev/ram0
```

```
TIMEOUT    100
```

-----<snip>-----

Now copy the initrdfs.gz and the kernel to /mnt and unmount /mnt. Now you execute:

```
syslinux -s /tmp/boot288.img
```

And detach the loop device:

```
losetup -d /dev/loop0
```

Voila, that's it.

1.3 Use grub as the bootloader

This recipe is heavily based on information provided to me by [Petr Konecny](#).

First make a 2.88MB zeroed image file.

```
dd if=/dev/zero of=/tmp/boot288.img bs=512 count=5760
```

Then set up the loop device.

```
losetup /dev/loop0 /tmp/boot288.img
```

Then format it with the ext2 filesystem.

```
mke2fs /dev/loop0
```

Mount it at /mnt.

```
mount /dev/loop0 /mnt
```

Create a directory named grub.

```
mkdir /mnt/grub
```

Copy the grub stage files onto the floppy image.

```
cp /usr/lib/grub/i386-pc/stage[12] /mnt/grub
```

Create a device map file.

```
echo "(fd0) /dev/loop0" > /tmp/dev.map
```

Install grub to the image.

```
/sbin/grub --device-map=/tmp/dev.map <<END
```

```
root (fd0)
```

```
setup (fd0)
```

```
END
```

-----<snip>-----

```
# grub menu file menu.lst
```

```
timeout                5
title                   Rescue System
root                    (fd0)
kernel                  /vmlinuz root=/dev/ram0 ramdisk_size=4096 init=/linuxrc rw
```

-----<snip>-----

Now copy the initrdfs.gz and the kernel to /mnt and unmount /mnt. Finally detach the loop device:

```
losetup -d /dev/loop0
```

Voila, that's it.

1.4 Use isolinux as the bootloader

Nothing easier than that.

First make a directory isolinux in the root tree of your rescue system:

```
mkdir $ROOTFS/isolinux
```

Copy the kernel image, the gzipped initrd image, isolinux.bin and isolinux.cfg to \$ROOTFS/isolinux

Make the cd image with the following command:

```
mkisofs -R -b isolinux/isolinux.bin -c isolinux/boot.cat -no-emul-boot -boot-load-size 4 -boot-info-table $ROOTFS > cd.iso
```

```
-----<snip>-----
```

```
# isolinux.cfg
```

```
DEFAULT RescueSystem
```

```
PROMPT 1
```

```
LABEL RescueSystem
```

```
KERNEL /isolinux/vmlinuz
```

```
APPEND initrd=initrdfs.gz ramdisk_size=40960 root=/dev/ram0
```

```
TIMEOUT 100
```

```
-----<snip>-----
```

Voila, that's it.

2. Multi boot cdrom images

You can use [isolinux](#) in conjunction with [memdisk](#) to create multi boot cds. You can also take a look at [nu2.nu](#), where you will find additional information on how to create multi boot cds.

The following recipe is based on information provided to me by Ralf Moll.

2.1 Tree hierarchy

You can download the tree shown below without the image files [here](#). Just untar the archive, copy the images you want to boot into the images directory and adjust the isolinux.cfg file.

```
cd_root/
|
|
|- RESCUECD
|
```

```

|- isolinux/
|
|   |- images/
|   |   |- dr_image.img
|   |   |- part_img.img
|   |   |- rescuecd.iso
|   |   |- win_95a.img
|   |   |- win_95b.img
|   |   |- win_98.img
|   |   |- win_98se.img
|   |   |- win_me.img
|   |
|   |- txt/
|   |   |- message.txt
|   |
|   |- initrdfs.gz
|   |- isolinux.bin
|   |- isolinux.cfg
|   |- memdisk
|   |- vmlinuz

```

2.2 isolinux.cfg

The rescuecd image is a special case 'cause we don't want to load the whole iso into RAM, so there is a special initrdfs.gz which takes care of starting it. The other images will all straightforward booted by the memdisk tool.

```

-----
DEFAULT 1
DISPLAY /isolinux/txt/message.txt
PROMPT 1

LABEL 1
  KERNEL /isolinux/vmlinuz
  APPEND initrd=initrdfs.gz init=/linuxrc ramdisk_size=5000 root=/dev/ram0
  TIMEOUT 100

LABEL 2
  KERNEL /isolinux/vmlinuz
  APPEND initrd=initrdfs.gz init=/linuxrc ramdisk_size=5000 root=/dev/ram0 cd_into_ram
  TIMEOUT 100

LABEL 3
  KERNEL /isolinux/memdisk
  APPEND initrd=images/dr_image.img
  TIMEOUT 100

LABEL 4
  KERNEL /isolinux/memdisk
  APPEND initrd=images/part_img.img
  TIMEOUT 100

```

```
LABEL 5
  KERNEL /isolinux/memdisk
  APPEND initrd=images/win_95a.img
  TIMEOUT 100
```

```
LABEL 6
  KERNEL /isolinux/memdisk
  APPEND initrd=images/win_95b.img
  TIMEOUT 100
```

```
LABEL 7
  KERNEL /isolinux/memdisk
  APPEND initrd=images/win_98.img
  TIMEOUT 100
```

```
LABEL 8
  KERNEL /isolinux/memdisk
  APPEND initrd=images/win_98se.img
  TIMEOUT 100
```

```
LABEL 9
  KERNEL /isolinux/memdisk
  APPEND initrd=images/win_me.img
  TIMEOUT 100
-----
```

2.3 Creating the image

```
mkisofs -R -b isolinux/isolinux.bin -c isolinux/boot.cat -no-emul-boot -boot-load-size 4 -boot-
info-table cd_root > rescuecd.iso
```

[homepage](#)

[top](#)

[Timo Benk](#) 24. October 2002

SOURCEFORGE.NET

