

# Remastering Knoppix

by

Cecil Watson

What is Knoppix?

KNOPPIX is a bootable CD with a collection of [GNU/Linux](#) software, automatic hardware detection, and support for many graphics cards, sound cards, SCSI and USB devices and other peripherals. KNOPPIX can be used as a Linux demo, educational CD, rescue system, or adapted and used as a platform for commercial software product demos. It is not necessary to install anything on a hard disk. Due to on-the-fly decompression, the CD can have up to 2 GB of executable software installed on it.

Why remaster?

1. Personalize distro to hardware.
2. Add/remove/update software.
3. Because you can! ;)

Examples of various remasters:

Auditor Security Collection is a Knoppix based LiveCD with focus on **Security**.

Feature: standardized working environment, remote support, optimal toolset.

Content: Tools for foot-printing, analysis, scanning, wireless driver, brute-forcing, cracking, information on standard configurations and password, 64 millions word list, productivity (web browsers, editor, graphic, report generator), Wellenreiter and Kismet with automatic hardware identification, wireless card drivers.

- Homepage: <http://www.remote-exploit.org>

Bioknoppix is a Knoppix based LiveCD from **Puerto Rico**, with focus on **molecular biology and bioinformatic**.

Content: EMBOSS 2.8.0, jemboss<sub>1</sub>, artemis, clustal, Cn3D, ImageJ, Biopython, Rasmol, Bioperl, Bioconductor

- Homepage: <http://bioknoppix.hpcf.upr.edu/>

The Bootable Cluster CD supports a drop-in solution for MPICH, LAM, PVM, [open Mosix](http://openmosix.sourceforge.net/) (<http://openmosix.sourceforge.net/>) **parallel** environments. Tools are available for debugging, compiling, and profiling parallel codes. Built from sources using GAR (a mix between BSD's "ports" system, Linux from scratch, and gentoo Linux).

- Homepage: <http://bccd.cs.uni.edu/>

DistccKNOPPIX is a LiveCD based on Damn Small Linux from **Canada**, running **distccd daemon**.

Feature: 4 versions of gcc, g++, cpp in single LiveCD. Run from RAM (toram) or hard disk (tohd) .

Content: Kernel 2.4.26, distcc 2.16 (port 3632), gcc (2.95.4, 3.2.3, 3.3.4, or 3.4.1), cpp (2.95.4, 3.2.3, 3.3.4, or 3.4.1), g++ (2.95.4, 3.2.3, 3.3.4, or 3.4.1).

- Homepage: <http://opendoorsoftware.com/cgi/http.pl?cookies=1&p=distccKNOPPIX>

[KnoppMyth](#) (my personal favorite ;) is a Knoppix based LiveCD with MythTV software.

Feature: transform PC into **set-top box** (record programs, rewind/pause live tv).

- Homepage: <http://mysettopbox.tv/knoppmyth.html>

You can find a complete list of available remasters at:

[http://www.knoppix.net/wiki/Knoppix\\_Customizations](http://www.knoppix.net/wiki/Knoppix_Customizations)

How to remaster?

Remastering consists of three simply stages:

1. Copying Knoppix to hard drive.
2. Chroot to source and make changes.
3. Rebuild ISO.

## **Copying Knoppix to hard drive:**

Although you can boot off the CD and build you remaster, my personal recommendation is to install Knoppix to the hard drive and running off the hard drive build your remaster.

1. Obtain a Knoppix CD and boot from the CD-ROM drive.
2. Install Knoppix to hard drive.
  - Create three partitions.
  - Minimum 3 gigs to install Knoppix.
  - 1 gig swap (mkswap /dev/hda2 && swapon /dev/hda2)
  - 5 gig partition in which to remaster.
  - Format /dev/hda3 with preferred filesystem.
  - Install.

After the installation is complete.

1. mount -o rw /dev/hda3 /mnt/hda3.
2. cd /mnt/hda3
3. mkdir -p source/KNOPPIX

4. cd source/KNOPPIX
5. wget http://debian.tu-bs.de/knoppix/debian/sarge-live-base/KNOPPIX.build.tar.gz
6. tar xzvf KNOPPIX.build.tar.gz
7. mkdir KNOPPIX.build/master
8. rsync -a --exclude "/KNOPPIX/KNOPPIX" /cdrom/KNOPPIX.build/master/
9. reboot

## Chroot to source and make changes:

1. Log in and su to root.
2. mount -o rw /dev/hda3 /mnt/hda3
3. cp /etc/resolv.conf /mnt/hda3/source/KNOPPIX/etc/dhclient/
4. chroot /mnt/hda3/source/KNOPPIX/
5. mount -t proc /proc proc
6. apt-get update (vi /etc/apt/sources.list if you need to)

Make your changes...

## Tips:

### \* **dpkg-query -l**

- If you want that list sorted by size (this way you can get rid of the biggies), type this:

### \* **dpkg-query -W --showformat='\${Installed-Size} \${Package}\n' | sort -n**

- To remove a package (and all packages dependant on it), type this:

### \* **apt-get remove --purge <name-of-package-to-remove>**

- To check for orphaned packages, type this:

### \* **deborphan**

- Want to save more space by getting rid of those pesky orphans (how cruel!), type this (**Warning**, you won't be prompted yes/no to remove these packages. When you press Enter after this command, those packages **will be gone**):

### \* **deborphan | xargs apt-get -y remove**

- If you're uncertain about the previous command and want to see what will happen without making any changes, just add the **-s** option to the apt-get command like this (you can do this with all of the apt-get commands, and it's a good habit to use this option

before mass operations like this one):

**\* deborphan | xargs apt-get -s -y remove**

- Now the good stuff. If you wish to add a package, type this:

**\* apt-get install <name-of-package-to-install>**

- What, don't know what packages to install? Type this. When the list appears, you can peruse (over 13k lines!) or search for things using **/<search-term>**:

**\* apt-cache search .\* | sort | less**

- When you're done removing and adding packages, a good way to clean up is by typing this:

**\* COLUMNS=200 dpkg -l |grep ^rc |awk '{print \$2}' | xargs dpkg -P**

- Also, because the Debian package system keeps a cache of downloaded packages, you may want to run the following to clear out those spare files:

**\* apt-get clean**

Working w/ X.

Outside of chroot:

```
cp /etc/X11/XF86Config-4 /mnt/hda3/source/KNOPPIX/etc/X11/
```

In chroot:

```
startx /usr/bin/startkde -- :1
```

You can move use **<CTRL><ALT><F7>** and **<CTRL><ALT><F8>** to switch between 'real' desktop and chroot desktop. Make changes. Once you are down working in X:

```
rm -fr /etc/X11/XF86Config-4
```

Making changes to the user 'knoppix'.

- cd /etc/
- tar -czf skel.tar.gz skel
- cd /home
- cp -R /etc/skel knoppix
- chown -R knoppix.knoppix knoppix/
- su knoppix
- cd
- make changes
- exit
- cd /home
- rm -fr /etc/skel
- mv knoppix /etc/skel
- chown -R root.root /etc/skel

To allow the user knoppix to use X as root (in chroot):

```
vi /etc/X11/Xwrapper.config
Change
allowed_users=console
to
allowed_users=anybody
```

Don't forget to change this back.

## Advance Tips:

Boot logo:

The boot logo is a 640\*400\*16 encoded image. It is located in: /KNOPPIX.build/master/boot/isolinux and called logo.16. To modify create an image and do the following as root:

1. pngtopnm <logo.png >logo.pnm
2. pmtolss16 <logo.pnm >logo.16
3. cp logo.16 /mnt/hda3/KNOPPIX.build/master/boot/isolinux/

Modifying Window Manager:

```
apt-get install window manager of choice
vi /etc/sysconfig/desktop
DESKTOP="window manager of choice"
Custom Kernel:
```

Get and build kernel source outside of chroot. The easier way to do this is with the Makefile in /usr/src. All additional kernel modules need to be recompiled for new kernel. At a minimum cloop needs to be recompiled. Reboot using new kernel.

1. Move and install kernel deb in chroot.
2. cd /KNOPPIX.build/master/boot/isolinux/
3. mkdir mroot
4. cp /boot/vmlinuz-version linux24 (or linux26)
5. gunzip minirt24.gz (or minirt26.gz)
6. mount -t ext2 -o loop minirt24 mroot
7. cd mroot/modules
8. cp /path/to/cloop.o (.ko if 2.6) cloop.o
9. cd scsi
10. replace all modules with new kernel modules
11. cd ../../..
12. umount mroot
13. rm -fr mroot
14. gzip -9 minirt24

## Rebuilding ISO:

In chroot:

1. cd /KNOPPIX.build/
  2. perl -pi -e 's/^(mount -o ro)/#\$1/' Knoppix-3.4.mkcompressed
  3. ./Knoppix-3.4.mkcompressed /KNOPPIX.build/master
- You'll see show output...

```
Update KNOPPIX version (3.7 2004-12-08) in file /
KNOPPIX.build/master/KNOPPIX/knoppix-version and /etc/knoppix-version
[N/y]? <ENTER>
Update Boot file(s) [N/y]?<ENTER>
Recreate KNOPPIX compressed FS [N/y]? y<ENTER>
```

Sweep/Clean system before remastering [N/y]? y<ENTER>

The script will do it's job!

Once it is ready for the next phase, you'll see:

Recreate KNOPPIX-DE isofile '/KNOPPIX.build/master.iso'? y<ENTER>

**\*\* Note \*\*** While it states -DE, it will build the ISO with whichever language you remastered for. Once complete you'll see:

Recreate KNOPPIX-EN isofile '/KNOPPIX.build/master-EN.iso'? n<ENTER>

BURN german CD version? n

4. umount /proc

5. exit

## Advance Tip:

If you have a cd burner:

1. cp /usr/bin/cdrecord .
2. Cdrecord -scanbus
3. vi Knoppix-3.4.mkcompressed

Search for CDRECORD

You'll find:

```
{ $CDRECORD dev=0,1,0 -atip 2>/dev/null | grep -qi "is
erasable"; } && BLANK="blank=fast"
$CDRECORD -v -dao -pad -eject gracetime=0 driveropts=burnfree
-overburn dev=0,1,0 speed=12 fs=24M $BLANK "$DEST.iso"
```

Change dev=0,1,0 to actual location of your burner. You can also change the speed. This this change, you can asked yes to burn a CD.

You can test the ISO by booting from the CD and using the follow:

```
knoppix bootfrom=/dev/hda3/source/KNOPPIX/KNOPPIX.build/master.iso
```

For this to work, must be running the same kernel on the boot CD as on the remaster. The filesystem that the ISO is on must be built into the kernel and not as a module.

Good luck and enjoy!

Cecil

References:

1. [http://www.knoppix.net/wiki/Knoppix\\_Remastering\\_Howto](http://www.knoppix.net/wiki/Knoppix_Remastering_Howto)
2. [http://www.knoppix.net/wiki/Knoppix\\_Customizations](http://www.knoppix.net/wiki/Knoppix_Customizations)
3. <http://www.stirnimann.com/mystuff/doc/knoppix.txt>
4. <http://gnubox.dyndns.org:8080/~sunil/knoppix.php>
5. Knoppix Hacks (<http://www.oreilly.com/catalog/knoppixhks/>)
6. A lot of head banging! :)