

Building a self-contained auto-configuring Linux system on an iso9660 filesystem or "Making Linux run from CD"

Klaus Knopper <knoppix@knopper.net>

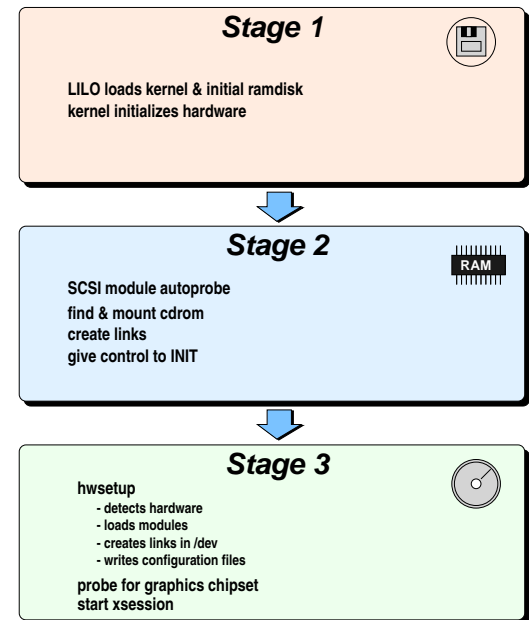
V2.1 – August 24, 2001

Abstract

Knoppix (Knopper's *nix) is an attempt to not only create a fully featured rescue/demo system on a single CD, but also to unburden the user from the task of hardware identification and configuration of drivers, devices and X11 for his or her specific hardware. The resulting product is supposed to be a platform CD with a stable GNU/Linux base system, that can be used to customize static installations for a specific purpose.

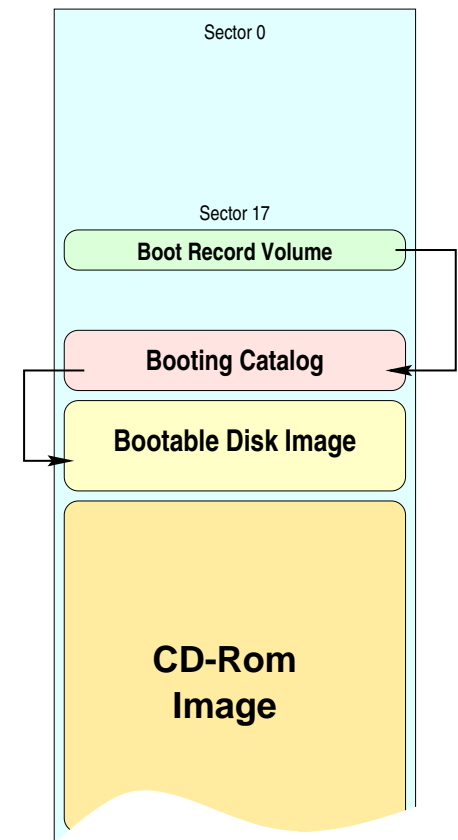
Table Of Contents

1. The KNOPPIX boot procedure
 - The El Torito Standard
 - The Bootfloppy
 - The Startup Scripts
2. The Filesystem
 - Chosing a filesystem
 - The compressed block device
 - The preparation scripts
3. Configuration and Applications
4. Planned Features



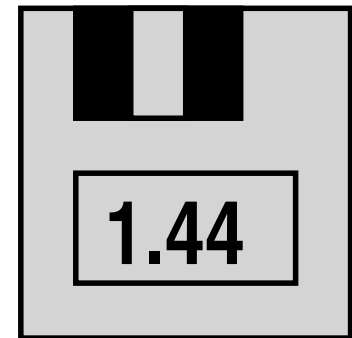
The El Torito Standard

- Standard for bootable CD-ROMs with boot index catalog for bootable images,
- Boot modes must be supported by (E)IDE or SCSI BIOS,
- Only allowed: 1,2, 1.44, 2.88 MB Floppy Image OR Harddisk emulation (BIOS access, not supported by standard Linux Kernel),
- Calibration and/or Geometry dependent problems when loading boot images via BIOS?



The Bootfloppy

- Standard 1.44 MB ext2 floppy image,
- Kernel (compressed) ca. 640K,
- 3MB initrd (SCSI modules, scripts, devices, shell) compressed ca. 790K,
- LILO stuff & load map ca. 10K.



The Startup Scripts

/linuxrc

- loads SCSI modules,
- checks for CD-Rom or harddisk partition with compressed image,
- mounts CD or harddisk partition with KNOPPIX installation,
- finds and uses existing swap partitions,
- creates additional ramdisk for **/home** and **/var**,
- gives control to **init**.

/etc/init.d/knoppix-autoconfig

- starts PCMCIA subsystem if chipset present,
- starts USB-Manager subsystem if chipset present,
- starts hardware detection and autoconfiguration (**hwsetup**),
- parses config files and modifies config files on ramdisk accordingly,
- creates KDE desktop icons and links,
- starts X-Server and xsession (with KDE, GNOME or TWM, depending on boot options and available memory).

Chosing a filesystem

- `/`, `/var`, `/home` (ramdisk) filesystems: `ext2` (read-write), configuration data, system directories, links
- `/cdrom` filesystem: `iso9660` (CD-Rom), Docs & Demos
- `/KNOPPIX` transparently decompressed: also `iso9660` (via `cloop`), read-optimized system and application filesystem (currently around 1.6 GB).

The compressed block device (1)

The compressed loopback device (cloop) was first introduced by Paul 'Rusty' Russel for the LinuxCare business card rescue CD.

- Read-only, transparently decompressing block device, based on `loopback.c`,
- block index permanently kept in system memory, uses kernel block buffering mechanism,
- filesystem-independent, no special libc with decompressing filesystem calls necessary,
- uses zlib/gzip compression, compression is done with user-space program **create_compressed_fs** in pipe mode (lots of virtual memory needed),
- CD-Rom friendly by reduced head movement (index/block caching, group read aka **block_read()** in `fs/block_dev.c`),
- Kernel 2.2 and 2.4 compatible with page-caching, aims to be SMP-safe.

The compressed block device (2)

Format of compressed block device file (all numbers and pointers in network byte order):

1. Header with small shell script as loader
(`insmod cloop.o file=/path/to/image`),
2. Header with information about *uncompressed* blocksize and # of blocks,
3. Block index:
 - (a) location of first compressed block in file,
 - (b) location of second compressed block in file,
 - (c) ...
 - (d) end-of-file location.
4. compressed block # 1,
5. compressed block # 2,
6. ...

The preparation scripts (1)

mkbootfloppy

```
MINIROOT=miniroot ; KERNEL=vmlinuz
FD=/dev/fd0H1440 ; FDDIR=/mnt/floppy
dd if=/dev/zero of=${MINIROOT}.img bs=4000k count=1
mke2fs -L "KNOPPIX Miniroot" -b 1024 -N 4096 -F \
    -q -m 0 ${MINIROOT}.img
mkdir -p initrd ; mount -t ext2 -o loop ${MINIROOT}.img initrd
rmdir initrd/lost+found
# Copy Programs and Devices to miniroot image
cp -a ${MINIROOT}/* initrd/ ; umount initrd
gzip -9cv miniroot.img > miniroot.img.gz
# Make ext2 bootfloppy
mke2fs -L "KNOPPIX Bootdisk" -F -q -i 262144 -N 8 -m 0 ${FD}
mount -t ext2 ${FD} ${FDDIR} ; rmdir ${FDDIR}/lost+found
cp -v boot.b ${KERNEL} ${MINIROOT}.img.gz ${FDDIR}/
# Install bootloader
lilo -v -v -C lilo-knoppix.conf ; umount ${FDDIR}
```

The preparation scripts (2)

mkcompressed

```
BLOCKSIZE=65536
SRC=/mnt/nfs/KNOPPIX ; DEST=/mnt/master/KNOPPIX/KNOPPIX

# Create compressed loopback file from NFS-mounted
# installation system
mkisofs -R -l -V "KNOPPIX iso9660 filesystem" \
        -hide-rr-moved -v $SRC | \
    nice -5 create_compressed_fs - $BLOCKSIZE > $DEST

# Make final CD Image
mkisofs -l -r -J -V "KNOPPIX V2.1" -hide-rr-moved -v \
        -b KNOPPIX/boot.img -c KNOPPIX/boot.cat \
        -o knoppix-cd.iso /mnt/master
```

Auto-Configuration

hwsetup uses the RedHat `kudzu`-Library to detect hardware, loads the necessary drivers and writes configuration data to files in `/etc/sysconfig/`.

```
int hw_setup(enum deviceClass dc, int verbose, int probeonly) {
    int i,mouse=0,cdrom=0,modem=0,scanner=0; struct device **currentDevs, *d;
    if((currentDevs=probeDevices(dc,BUS_UNSPEC,PROBE_ALL))==NULL) return -1;
    for(i=0;(d=currentDevs[i]);i++) {
        if(verbose) hw_info(d);
        if(!probeonly) {
            if(d->bus==BUS_ISAPNP&&configure_isapnp(d,verbose)) continue;
            switch(d->class) {
                case CLASS_MOUSE: if(!mouse) writeconfig("/etc/sysconfig/mouse",d,verbose);
                                    link_dev(d,"mouse", mouse++,verbose); break;
                case CLASS_CDROM: link_dev(d,"cdrom", cdrom++,verbose); break;
                case CLASS_MODEM: link_dev(d,"modem", modem++,verbose); break;
                case CLASS_SCANNER: link_dev(d,"scanner",scanner++,verbose); break;
                case CLASS_VIDEO: writeconfig("/etc/sysconfig/xserver",d,verbose);
                                    break;
                case CLASS_AUDIO: writeconfig("/etc/sysconfig/sound",d,verbose);
                                    load_mod(d->driver,verbose); break;
                case CLASS_NETWORK: writeconfig("/etc/sysconfig/netcard",d,verbose);
                                    load_mod(d->driver,verbose); break;
                ...
            }
        }
    }
}
```

Preinstalled software

- Up-to-date Kernel 2.4,
- Full-featured Debian/testing (Non-US version) base system,
- XFree 4.1 + XFree 3.3.6 (for older graphics cards),
- ReiserFS utilities & standard recovery tools,
- KDE 2.2-final,
- GIMP 1.2 with scanner support, KOffice,
- xmms 1.2.3 w/ mp3,ogg+smpeg plugins,
- openssl, openssh, konqueror/SSL,
- Nessus, Ethereal, ettercap (Network & security checks),
- Netscape™ 4.78, Star-/OpenOffice™ 5.2,
- Demos & Games.

Current Features

- Rescue System, recovery from different filesystems (including NTFS),
- full-featured working Linux application platform, internet-ready,
- platform for custom software, Product Demos, Multimedia collections, Games, Presentations, ...
- Tools for network administrators and security experts,
- firewall/routing/proxy software,
- can be customized as Installer or Launcher of other operating systems,
- file, print and information services (NFS, CUPS, SAMBA, MySQL, apache, ...).

Planned Features

- Split bootfloppy in 2x 1.44MB (?),
- More (and newer) graphics cards support (XFree 4.0?),
- Fancy configuration frontend for "expert" mode,
- Easier customization procedure with lean base system.

References & Contact

El Torito Standard:

<http://www.ptltd.com/products/specs-cdrom.pdf>

Linux-Bootable-Businesscards Project:

<http://http://www.lnx-bbc.org/>

Nessus Security Network Scanner:

<http://www.nessus.org/>

KDE Project Homepage:

<http://www.kde.org/>

GNOME Project Homepage:

<http://www.gnome.org/>

KNOPPIX CD / Sources / Contact Address:

<http://www.knopper.net/knoppix/>

<mailto:knoppix@knopper.net>