

Creating CD Images: mkisofs & dd

Imag(in)e your hard disk

The commands `mkisofs` and `dd` can be used to create images from files on your hard disk, or from a CD, which you can test mount before copying to a new CD.

BY HEIKE JURZIK

An image maps a file system to a single file. Data on CD-ROMs are organized in a special file system that differs from the file system used by a hard disk, so you cannot simply copy individual files to the CD.

Extracting a CD Image

The abbreviation `dd` actually stands for “convert and copy” – but `cc` had already been assigned to the C compiler. The tool transfers data between various storage media, performing conversion operations, when required. `dd` does not simply copy individual files, but can access devices directly. This allows you to copy complete hard disks, partitions and of course, CD-ROMs.

Options are entered without a minus sign. You can specify the **block size** for the input and output files, and the number of blocks to copy. You can issue the following command to save the boot sector on your first hard disk, for example:

```
asteroid:~# dd if=/dev/hda 2
of=bootsector bs=512 count=1
```

The `if` (“infile”) option specifies the input file (this defaults to standard input); `of` (“outfile”) specifies the target file. The `bs` (“blocksize”) parameter allows you to supply the block size; `count` specifies the number of blocks to copy. So the above command translates to “copy from the first IDE disk

(`/dev/hda`) to a file called `bootsector` the first block (`count = 1`), which comprises 512 bytes (`bs = 512`).“

Another typical area of application is when creating a boot floppy, for example, by copying the kernel (`vmlinuz`) to a floppy disk:

```
asteroid:~# dd if=/vmlinuz 2
of=/dev/fd0
```

The floppy disk created by this process would allow you to boot your Linux system, despite a Windows installation having overwritten the boot sector. The simplest way to copy a data CD is to use the following:

```
asteroid:~# dd if=/dev/cdrom 2
of=bla.iso
1349272+0 records in
1349272+0 records out
```

You do not need to mount the CD to do so – `dd` works independently of the type

of data on the CD, the exception being audio CDs – but you can always use another utility, such as `cdda2wav` [2], for this purpose.

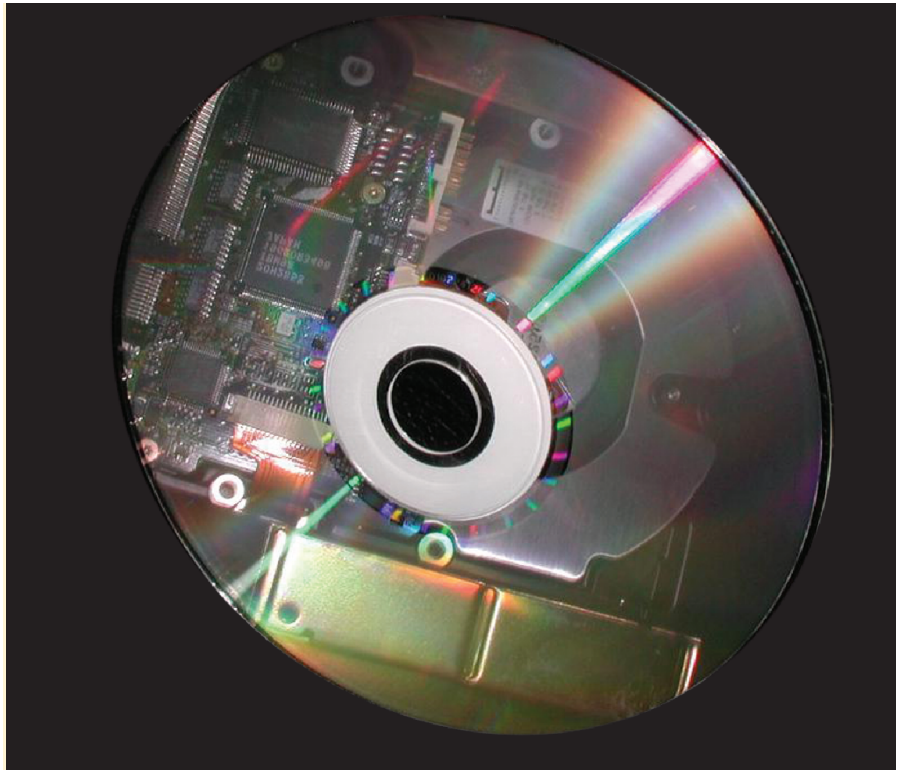
Imagine!

If you want to collect data before writing it to a CD, you will need to create an **ISO** image, and this is where the `mkisofs` (“make ISO9660 filesystem” [3]) command comes in useful. The program creates a file based on the ISO9660 guidelines.

The following command copies files from a directory to an image:

```
mkisofs [parameter] -o file.iso 2
/tmp/data
```

The `-o` option specifies the output file where the image will be stored (in this case `file.iso`); the source is `/tmp/data`. The most common parameters are `-r` and `-J` – **Rockridge** and **Joliet** extensions – expect to use them often. This ensures



that your CD can be read by a Windows operating system. Instead of `-r` you can also specify `-R` here. This parameter uses Rockridge, but retains the privileges and file ownerships – this is the option to go for when creating a backup of the `/home` directory on your own workstation.

Bootable CDs

The `-b` option is also useful, as it allows you to create a bootable CD. The file size of the file specified by `-b` must be 1.2 MB, 1.44 MB or 2.88 MB and the file must contain the image of a bootable floppy – of course the boot sector must be stored in the first 512 bytes. You might additionally like to specify the `-c` option, which indicates the path to a boot catalog file to be created relative to the root directory of the CD, for example, `mkisofs -c images/boot.cat`.

You can use the `-V` option to set the VolumeID, that is the “name” of the CD, e.g. `-V “Backup for December 1 2002”`. Typical Linux Magazine subscription CD ROM images are created by using the following command:

```
mkisofs -V "LU01-2003" -J -r -P 2
"Linux New Media" -b Boot/2
cdrom.img -c Boot/boot.cat 2
-o lu-2003-01.iso lu01/
```

Isolinux

The usefulness of a 1.44 MB or 2.88 MB boot image is obviously somewhat restricted – if you want to store both the kernel and some modules in the image, you will soon run out of space. The Isolinux system, an extension of the SysLinux floppy boot loader [1], provides an useful alternative. You can use a configu-

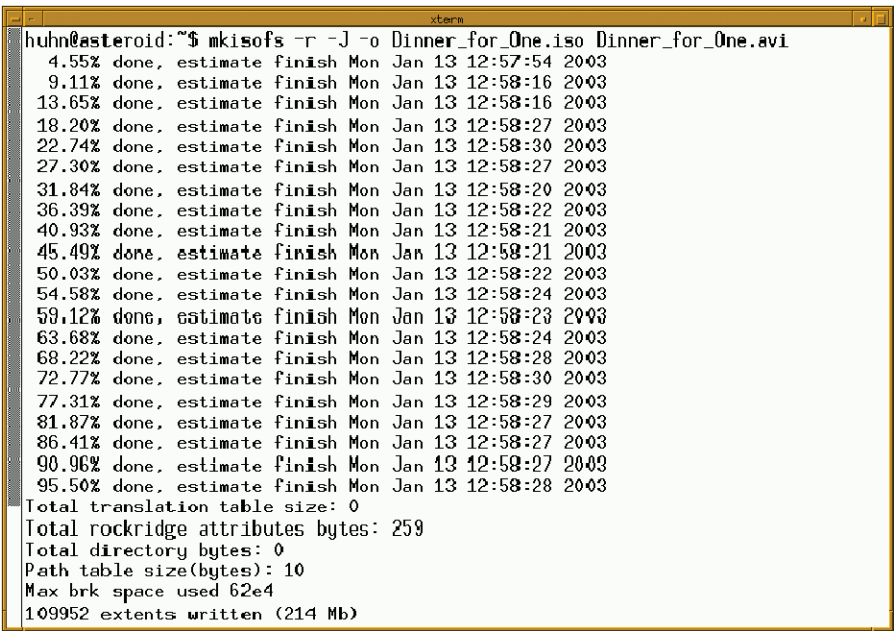


Figure 1: `mkisofs` keeps you informed

ration file to activate an arbitrary kernel stored in the CD file system. It is also possible to load RAM disks required at Linux startup directly from the CD. The Isolinux boot menu is configurable via the `isolinux/islinux.cfg` file, which contains sections similar to the following (like the LILO configuration file)

```
label linux
kernel vmlinuz
append initrd=disk.rdz ramdisk_2
size=32000 root=/dev/ram3 vga2
=788 automatic=method:cdrom
```

Both the kernel `vmlinuz` and the RAM disk `disk.rdz` are stored in the `/isolinux` directory on the CD-ROM; subdirectories below `/isolinux` can also be used.

The following `mkisofs` syntax creates a CD image with an Isolinux boot loader:

```
mkisofs -J -r -b isolinux/iso2
linux.bin -c isolinux/boot.cat 2
-no-emul-boot -boot-load-size 42
-boot-info-table 2
-o output.iso directory/
```

On the Safe Side

Before you use your favorite CD creator (such as `cdrecord` or `X-CD-Roast`) to burn the new image on a CD, you might like to ensure that the image is error free. The `mount` command not only allows you to access storage media such as hard disks or CDs into your Linux filesystem, but also to mount ISO images just like any other devices. The admin user `root` can type the following command to mount an image of this type:

```
asteroid:~# mount -o loop 2
foo.iso /mnt/tmp/
```

The command may need some help recognizing the filesystem; in this case, simply supply an additional parameter, `-t iso9660`. The image can be unmounted by typing `umount /mnt/tmp` on the command line.

GLOSSARY

Block size: A block is a contiguous collection of bytes on a storage medium. Some devices (appropriately known as “block devices”) organize their data in blocks, in contrast to “character devices”, which perform character oriented operations and read or write individual bytes.

ISO: Linux supports various types of file-system. These include both native filesystems, such as `ext2`, `ext3` or `reiserfs`, and the `vfat` and `ntfs` filesystems used by Windows. ISO 9660, to quote the full name, is the standard for file management on CD ROMs. Although only short file names were originally supported,

the Rockridge and Joliet extensions removed this restriction.

Rockridge: The Rockridge Extensions extend the ISO CD filesystem to support typical UNIX file information for the owner and group, access privileges (read, write, execute) and symbolic links. This allows data to be copied from a UNIX filesystem to a CD without any loss of information. Rockridge additionally supports longer file names.

Joliet: An alternative extension of the ISO9660 filesystem by Microsoft and named Joliet. Joliet also supports long file names.

INFO

- [1] <http://syslinux.zytor.com/iso.php>
- [2] <http://www.cddazwav.de/>
- [3] <http://www.tldp.org/HOWTO/Filesystems-HOWTO-8.html>