Multimedia in Linux

Alon Altman

alon@haifux.org

Haifa Linux Club

Playing Media Files

Sound Systems — Kernel

There are two sound systems built into the Linux kernel:

- OSS (Open Sound System) — The older digital audio system since the 2.0.x kernels.
- ALSA (Advanced Linux Sound Architecture) — The newer sound system being introduced in 2.6. Supports more sound cards and allows multiple streams at once.

Sound Systems — Multiplexers

In addition, there are sound mixers out of the kernel that are in use. These programs capture the OSS `/dev/dsp` device and multiplex sounds played through them. The common ones are:

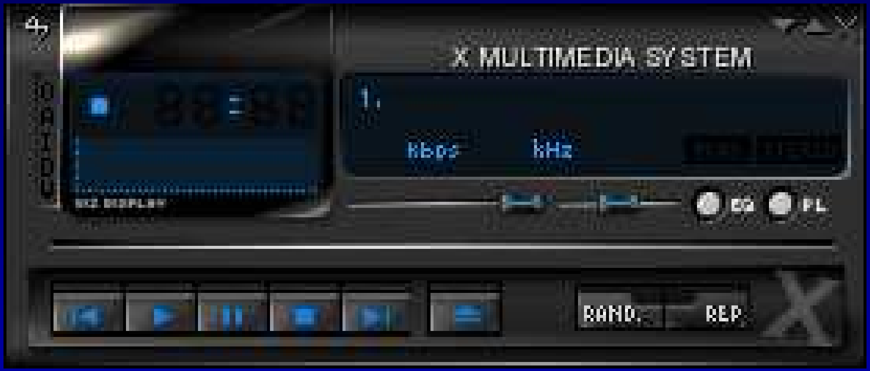
- EsounD — The Enlightened Sound Daemon, used mostly by gnome.
- aRts — Analog Realtime Synthesizer, used by KDE.

Use `esddsp` and `artsdsp` to simulate standard OSS when using these daemons. For example:

```
#artsdsp realplay mymovie.rm
```

Playing Music — xmms

The most common music-playing software is `xmms`, which allows you to play many media files including MP3 and Ogg Vorbis. `xmms` supports EsounD and aRts for playback in addition to OSS and ALSA.



RedHat and Fedora users have a crippled version of `xmms` without the MP3 plugin, and therefore should download and install the plugin from:

```
http://dag.wieers.com/packages/xmms-mp3/
```

Playing movies – MPlayer

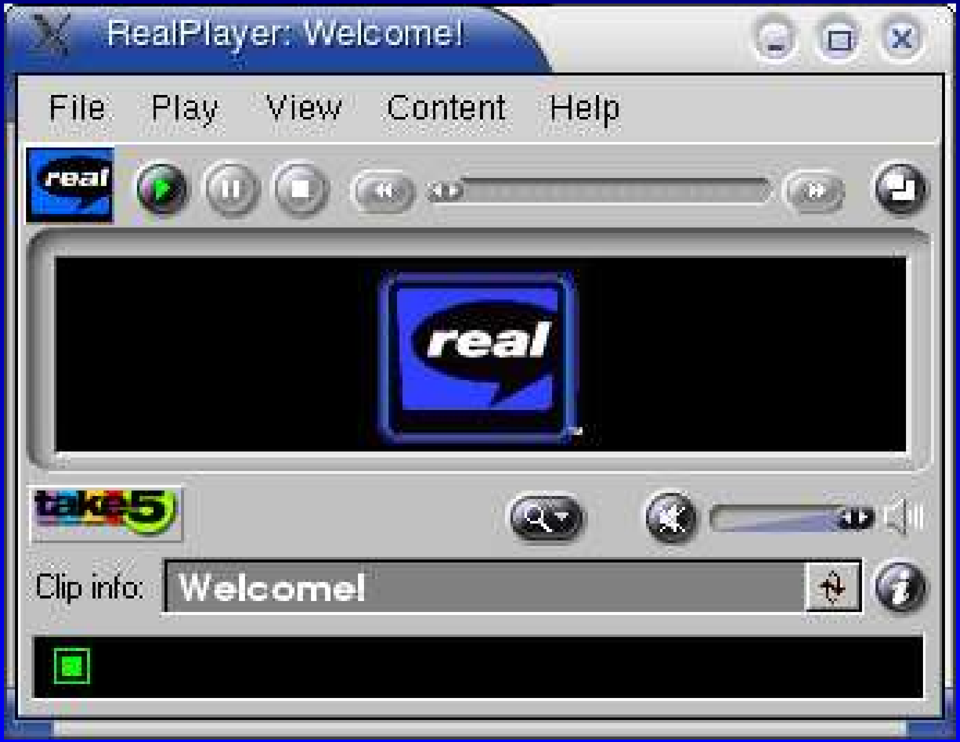
MPlayer is *the* movie player for Linux. It's GUI is still limited, so it's recommended to start MPlayer from the command line.

Due to the dynamic nature of MPlayer, it is recommended to download and compile MPlayer from source. You should also install the binary codecs available from the MPlayer homepage and additional codecs as needed.

To check what codecs are to be used, look at the output of `./configure`. If want to install additional drivers, check the MPlayer documentation on how to install them, then rerun `./configure` and make sure the codecs are enabled. Finally run `make` and `make install` as usual.

RealPlayer

To install the notorious RealPlayer (closed source), download it from the “community supported” site at: `http://forms.real.com/real/player/unix/unix.html`. Make sure you also download the linked codec upgrade package to support newer RealPlayer 9 codecs.



Mozilla Plugins

- Macromedia Flash support in Linux is available only via a binary plugin supplied by Macromedia at `http://www.flash.com/`. Simply download the plugin and follow the installation instructions. Make sure that the flash plugin is in your Mozilla plugins directory.
- The java plugin can be downloaded from `http://www.java.com/`. After installing, create a symbolic link from the java plugin from: `/usr/java/j2re1.4.2_03/plugin/i386/ns610-gcc32/` to your mozilla plugins directory: `/usr/lib/mozilla/plugins/`
- `plugger` allows seamless integration of various plugins with Mozilla. Get it from `http://fredrik.hubbe.net/plugger.html`

Burning CDs

Using a GUI

Burning CDs under Linux today is as easy as running `xcdroast` as root and using a GUI, assuming your burner is properly configured.

The CD Writing Process

- Prepare your system for writing CDs (done only once)
- Prepare source directory/wav files
- Make the ISO image.
- Write the ISO image to a CD.

Preparation

- Install your CD Writer Drive physically. **Note:** Make sure your CD Writer drive is *not* on the same IDE cable as your HD with the CD-images, to prevent buffer underruns.
- Install kernel modules `ide-scsi` (for IDE only), `sg` and `sr_mod`. You can either re-compile the kernel, or insmod the modules.
- **For IDE only** - edit `/etc/lilo.conf` and add in your configuration a line `append="hdd=ide-scsi"`, where hdd is the drive of your CD-Writer.
- Reboot your computer. Run as root: `cdrecord --scanbus` and make sure you see your cd-writer listed.

Preparation of Source Directory

- In order to write CDs you need the source data be put in one directory (actually, the newest version of `mkisofs` knows to handle multiple source dirs). If your files are scattered all over the hard disk, you can either copy the files to one source directory, or use symbolic links, and create an image directory that includes only symbolic links to the source files.
- For writing CD-Audio tracks, you need to prepare WAV files that will be converted to CD-Audio.

Running mkisofs

`mkisofs` is the program that converts files on your filesystem to a CD image. The syntax is:

```
mkisofs [options] -o filename pathspe
```

- The `-o` option specifies the name output *image file* to be written.
- The path specification is the name of the source directory.
- To use multiple source directories scattered arbitrarily on the filesystem, you may use the `-graft-points` option. `man mkisofs` for details.

mkisofs options

`-v` Enable verbose mode - always use this!

`-r` Enable Rock Ridge (UNIX permissions) - for backups.

`-J` Enable Joliet (Windows-Readable LFNs)

`-jcharset` Specify charset for file names in Windows. Use charset `iso8859-8` for standard Hebrew support.

`-f` Follow symbolic links - Important if you’ve symlinked your source directory.

`-l` Allow long file names (31 characters)

`-L` Allow filenames to begin with a period.

Refer to `man mkisofs` for complete list of options.

Testing the CD-Image

- Linux lets you `mount` a CD-Image as if it is a real CD.
- This is by using the special `loop` device in the kernel.
- To test a cdrom image issue the command:

  ```
  # mount -o loop,ro -t iso9960
              image.iso /mnt/cdimage
  ```

- Don't forget to unmount the image after testing:

  ```
  # umount /mnt/cdimage
  ```

Running cdrecord

`cdrecord` is program that actually writes the disk. The syntax is:

```
cdrecord [general options]
        dev=device [track options]
        track1...trackn
```

’device’ is the SCSI id of the device, in the format scsibus,target,lun. This data can be found by running `cdrecord --scanbus`. For IDE drives, this usually is 0,0,0.

cdrecord general options

The most important options include:

`-v`	Enable verbose mode - always use this!
`-dummy`	Simulation mode. Try this before real writing.
`-eject`	Eject after write.
`speed=#`	Set writing speed.
`fs=#`	Set FIFO size (buffer). Default is 4MB. Mostly used when piping data directly from `mkisofs`.
`-dao`	Disk-At-Once mode, eliminates silence between audio tracks.

cdrecord track options

`-audio` All subsequent tracks are audio tracks (au or wav files).

`-data` All subsequent tracks are data tracks (ISO images).

`-pad` Pad data/audio tracks to frame. Needed when writing audio.

Refer to `man cdrecord` for detailed explanation of all options.

Writing BIN/CUE images

Sometimes, CD images are supplied in a BIN/CUE format and not ISO. This format is used to specify the contents of an entire disc as opposed to just one track. Therefore, the BIN/CUE format is much more flexible and allows for creation of *any* type of CD.
To write these raw images, we use a different CD-Writing tool called `cdrdao`:

```
cdrdao  write --device 0,0,0
        [--simulate] -v 2 --buffers
        64 --speed speed image.cue
```

Note: The BIN file must be in the same directory as the CUE file for this to work.

Converting BIN/CUE to ISO

There are things that cannot be done with a BIN/CUE image. For example, loopback mounting of the image file and burning the image alongside other audio (or data) tracks.

In these special cases, you can use `bchunk` to extract the ISO (=data) and WAV (=audio) tracks from the BIN and CUE combination.

Usage: `bchunk -w` *`image.bin`* *`image.cue`* *`basename`*

This generates files with the given basename with the data and audio tracks of the CD