The Linux Sound HOWTO

Table of Contents

The Linux Sound HOWTO......1
Jeff Tranter, tranter@pobox.com......1
1.Introduction......1
2.Sound Card Technology......1
3.Supported Hardware......1
4.Installation......1
5.Applications Supporting Sound......2
6.Answers To Frequently Asked Questions......2
7.References......3
1.Introduction......3
1.1 Acknowledgments......3
1.2 New versions of this document......3
1.3 Feedback......4
1.4 Distribution Policy......4
2.Sound Card Technology......4
3.Supported Hardware......5
3.1 Sound Cards......6
3.2 Alternate Sound Drivers......8
3.3 PC Speaker......10
3.4 Parallel Port......10
4.Installation......10
4.1 Installing the Sound Card......11
4.2 Configuring Plug and Play......11
4.3 Configuring the Kernel......12
4.4 Creating the Device Files......12
4.5 Booting Linux and Testing the Installation......13
4.6 Troubleshooting......15
Step 1: Make sure you are really running the kernel you compiled......15
Step 2: Make sure the kernel sound drivers are compiled in......16
Step 3: Did the kernel detect your sound card during booting?......16
Step 4: Can you read data from the dsp device?......17
When All Else Fails......17
5.Applications Supporting Sound......17
6.Answers To Frequently Asked Questions......18
6.1 What are the various sound device files?......18
6.2 How can I play a sound sample?......20
6.3 How can I record a sample?......20
6.4 Can I have more than one sound card?......20
6.5 Error: No such file or directory for sound devices......21
6.6 Error: No such device for sound devices......21
6.7 Error: No space left on device for sound devices......21
6.8 Error: Device busy for sound devices......22
6.9 I still get device busy errors!......22
6.10 Partial playback of digitized sound file......22
6.11 There are pauses when playing MOD files......23
6.12 Compile errors when compiling sound applications......23
6.13 SEGV when running sound binaries that worked previously......23

Table of Contents

6.14 What known bugs or limitations are there in the sound driver?....23
6.15 Where are the sound driver ioctls() etc. documented?....24
6.16 What CPU resources are needed to play or record without pauses?....24
6.17 Problems with a PAS16 and an Adaptec 1542 SCSI host adaptor....24
6.18 Is it possible to read and write samples simultaneously?....25
6.19 My SB16 is set to IRQ 2, but configure does not allow this value....26
6.20 If I run Linux, then boot DOS, I get errors and/or sound applications do not work properly....26
6.21 Problems running DOOM under Linux....26
6.22 How can I reduce noise picked up by my sound card?....27
6.23 I can play sounds, but not record....27
6.24 My "compatible" sound card only works if I first initialize under MS−DOS....27
6.25 My 16−bit SoundBlaster "compatible" sound card only works in 8−bit mode under Linux....28
6.26 Where can I find sound applications for Linux?....28
6.27 Can the sound driver be compiled as a loadable module?....28
6.28 Can I use a sound card to replace the system console beep?....28
6.29 What is VoxWare?....29
6.30 Sox/Play/Vplay reports "invalid block size 1024"....29
6.31 The mixer settings are reset whenever I load the sound driver module....29
6.32 Only user root can record sound....30
6.33 Is the sound hardware on the IBM ThinkPad supported?....30
6.34 Applications fail because my sound card has no mixer....30
6.35 Problems with a SB16 CT4170....30
6.36 How to connect a MIDI keyboard to a soundcard....31
6.37 Problems with IRQ 15 and Ensoniq PCI 128....32
6.38 Where can I get freely available MIDI patches to run SoftOSS?....33
7.References....33

The Linux Sound HOWTO

Jeff Tranter, tranter@pobox.com

v1.20, 24 March 1999

This document describes sound support for Linux. It lists the supported sound hardware, describes how to configure the kernel drivers, and answers frequently asked questions. The intent is to bring new users up to speed more quickly and reduce the amount of traffic in the Usenet news groups and mailing lists.

1.Introduction

- 1.1 Acknowledgments
- 1.2 New versions of this document
- 1.3 Feedback
- 1.4 Distribution Policy

2.Sound Card Technology

3.Supported Hardware

- 3.1 Sound Cards
- 3.2 Alternate Sound Drivers
- 3.3 PC Speaker
- 3.4 Parallel Port

4.Installation

- 4.1 Installing the Sound Card
- 4.2 Configuring Plug and Play
- 4.3 Configuring the Kernel
- 4.4 Creating the Device Files
- 4.5 Booting Linux and Testing the Installation
- 4.6 Troubleshooting

5.Applications Supporting Sound

6.Answers To Frequently Asked Questions

- 6.1 What are the various sound device files?
- 6.2 How can I play a sound sample?
- 6.3 How can I record a sample?
- 6.4 Can I have more than one sound card?
- 6.5 Error: No such file or directory for sound devices
- 6.6 Error: No such device for sound devices
- 6.7 Error: No space left on device for sound devices
- 6.8 Error: Device busy for sound devices
- 6.9 I still get device busy errors!
- 6.10 Partial playback of digitized sound file
- 6.11 There are pauses when playing MOD files
- 6.12 Compile errors when compiling sound applications
- 6.13 SEGV when running sound binaries that worked previously
- 6.14 What known bugs or limitations are there in the sound driver?
- 6.15 Where are the sound driver ioctls() etc. documented?
- 6.16 What CPU resources are needed to play or record without pauses?
- 6.17 Problems with a PAS16 and an Adaptec 1542 SCSI host adaptor
- 6.18 Is it possible to read and write samples simultaneously?
- 6.19 My SB16 is set to IRQ 2, but configure does not allow this value.
- 6.20 If I run Linux, then boot DOS, I get errors and/or sound applications do not work properly.
- 6.21 Problems running DOOM under Linux
- 6.22 How can I reduce noise picked up by my sound card?
- 6.23 I can play sounds, but not record.
- 6.24 My "compatible" sound card only works if I first initialize under MS−DOS.
- 6.25 My 16−bit SoundBlaster "compatible" sound card only works in 8−bit mode under Linux.
- 6.26 Where can I find sound applications for Linux?
- 6.27 Can the sound driver be compiled as a loadable module?
- 6.28 Can I use a sound card to replace the system console beep?
- 6.29 What is VoxWare?
- 6.30 Sox/Play/Vplay reports "invalid block size 1024"
- 6.31 The mixer settings are reset whenever I load the sound driver module
- 6.32 Only user root can record sound
- 6.33 Is the sound hardware on the IBM ThinkPad supported?
- 6.34 Applications fail because my sound card has no mixer
- 6.35 Problems with a SB16 CT4170
- 6.36 How to connect a MIDI keyboard to a soundcard
- 6.37 Problems with IRQ 15 and Ensoniq PCI 128
- 6.38 Where can I get freely available MIDI patches to run SoftOSS?

7.References

1.Introduction

This is the Linux Sound HOWTO. It is intended as a quick reference covering everything you need to know to install and configure sound support under Linux. Frequently asked questions about sound under Linux are answered, and references are given to some other sources of information on a variety of topics related to computer generated sound and music.

The scope is limited to the aspects of sound cards pertaining to Linux. See the other documents listed in the *References* section for more general information on sound cards and computer sound and music generation.

1.1 Acknowledgments

Much of this information came from the documentation provided with the sound driver source code, by Hannu Savolainen (hannu@opensound.com). Thanks go to Hannu, Alan Cox, and the many other people who developed the Linux kernel sound drivers and utilities.

Thanks to the SGML Tools package, this HOWTO is available in several formats, all generated from a common source file.

1.2 New versions of this document

New versions of this document will be periodically posted to the comp.os.linux.answers newsgroup. They will also be uploaded to various anonymous ftp sites that archive such information including ftp://metalab.unc.edu/pub/Linux/docs/HOWTO/.

Hypertext versions of this and other Linux HOWTOs are available on many world−wide web sites, including http://metalab.unc.edu/LDP/. Most Linux CD−ROM distributions include the HOWTOs, often under the /usr/doc directory, and you can also buy printed copies from several vendors. Sometimes the HOWTOs available from CD−ROM vendors, ftp sites, and printed format are out of date. If the date on this HOWTO is more than six months in the past, then a newer copy is probably available on the Internet.

Please note that, due to the dynamic nature of the Internet, all web and ftp links listed in this document are subject to change.

Translations of this document are available in several languages:

Chinese: http://www.linux.org.tw/CLDP/Sound−HOWTO.html

French: http://www.freenix.org/unix/linux/HOWTO/

Japanese: http://yebisu.ics.es.osaka−u.ac.jp/linux/

Korean: http://kldp.linux−kr.org/HOWTO/html/Sound/Sound−HOWTO.html

Russian: http://www.phtd.tpu.edu.ru/~ott/russian/linux/howto−rus/Sound−HOWTO.html

Spanish: ftp://ftp.insflug.org/es

Most translations of this and other Linux HOWTOs can also be found at http://metalab.unc.edu/pub/Linux/docs/HOWTO/translations/ and ftp://metalab.unc.edu/pub/Linux/docs/HOWTO/translations/. If you make a translation of this document into another language, let me know and I'll include a reference to it here.

1.3 Feedback

I rely on you, the reader, to make this HOWTO useful. If you have any suggestions, corrections, or comments, please send them to me, tranter@pobox.com, and I will try to incorporate them in the next revision.

I am also willing to answer general questions on sound cards under Linux, as best I can. Before doing so, please read all of the information in this HOWTO, and send me detailed information about the problem. Please do not ask me about using sound cards under operating systems other than Linux.

If you publish this document on a CD−ROM or in hardcopy form, a complimentary copy would be appreciated. Mail me for my postal address. Also consider making a donation to the Linux Documentation Project to help support free documentation for Linux. Contact the Linux HOWTO coordinator, Tim Bynum mailto:linux−howto@metalab.unc.edu, for more information.

1.4 Distribution Policy

2.Sound Card Technology

This section gives a *very* cursory overview of computer audio technology, in order to help you understand the concepts used later in the document. You should consult a book on digital audio or digital signal processing in order to learn more.

Sound is an *analog* property; it can take on any value over a continuous range. Computers are *digital*; they like to work with discrete values. Sound cards use a device known as an *Analog to Digital Converter* (A/D or ADC) to convert voltages corresponding to analog sound waves into digital or numeric values which can be stored in memory. Similarly, a *Digital to Analog Converter* (D/A or DAC) converts numeric values back to an analog voltage which can in turn drive a loudspeaker, producing sound.

The process of analog to digital conversion, known as sampling, introduces some error. Two factors are key in determining how well the sampled signal represents the original. *Sampling rate* is the number of samples made per unit of time (usually expresses as samples per second or Hertz). A low sampling rate will provide a less accurate representation of the analog signal. Sample size is the range of values used to represent each sample, usually expressed in bits. The larger the sample size, the more accurate the digitized signal will be.

Sound cards commonly use 8 or 16 bit samples at sampling rates from about 4000 to 44,000 samples per second. The samples may also be contain one channel (mono) or two (stereo).

FM Synthesis is an older technique for producing sound. It is based on combining different waveforms (e.g. sine, triangle, square). FM synthesis is simpler to implement in hardware that D/A conversion, but is more difficult to program and less flexible. Many sound cards provide FM synthesis for backward compatibility with older cards and software. Several independent sound generators or *voices* are usually provided.

Wavetable Synthesis combines the flexibility of D/A conversion with the multiple channel capability of FM synthesis. With this scheme digitized voices can be downloaded into dedicated memory, and then played, combined, and modified with little CPU overhead. State of the art sound cards all support wavetable synthesis.

Most sound cards provide the capability of *mixing*, combining signals from different input sources and controlling gain levels.

MIDI stands for Musical Instrument Digital Interface, and is a standard hardware and software protocol for allowing musical instruments to communicate with each other. The events sent over a MIDI bus can also be stored as MIDI files for later editing and playback. Many sound cards provide a MIDI interface. Those that do not can still play MIDI files using the on−board capabilities of the sound card.

MOD files are a common format for computer generated songs. As well as information about the musical notes to be played, the files contain digitized samples for the instruments (or voices). MOD files originated on the Amiga computer, but can be played on other systems, including Linux, with suitable software.

3.Supported Hardware

This section lists the sound cards and interfaces that are currently supported under Linux. The information here is based on the latest Linux kernel, which at time of writing was version 2.2.4. This document only applies to the sound drivers included with the standard Linux kernel source distribution. There are other sound drivers available for Linux (see the later section entitled Alternate Sound Drivers).

For the latest information on supported sound cards and features see the files included with the Linux kernel source code, usually installed in the directory /usr/src/linux/Documentation/sound.

The information in this HOWTO is valid for Linux on the Intel platform.

The sound driver should also work with most sound cards on the Alpha platform. However, some cards may conflict with I/O ports of other devices on Alpha systems even though they work perfectly on i386 machines, so in general it's not possible to tell if a given card will work or not without actually trying it.

Users have reported that the sound driver was not yet working on the PowerPC version of Linux, but it should be supported in future.

Sound can be configured into the kernel under the MIPs port of Linux, and some MIPs machines have EISA slots and/or built in sound hardware. I'm told the Linux−MIPs group is interested in adding sound support in the future.

The Linux kernel includes a separate driver for the Atari and Amiga versions of Linux that implements a compatible subset of the sound driver on the Intel platform using the built−in sound hardware on these machines.

The SPARC port of Linux currently has sound support for some models of Sun workstations. I've been told that the on−board sound hardware works but the external DSP audio box is not supported because Sun has not released the specifications for it.

3.1 Sound Cards

The following sound cards are supported by the Linux kernel sound driver. Some of the items listed are audio chip sets rather than models of sound cards. The list is incomplete because there are many sound cards compatible with these that will work under Linux. To add further to the confusion, some manufacturers periodically change the design of their cards causing incompatibilities and continue to sell them as the same model.

- 6850 UART MIDI Interface
- AD1816/AD1816A based cards
- ADSP−2115
- ALS−007 based cards (Avance Logic)
- ATI Stereo F/X (no longer manufactured)
- Acer FX−3D
- AdLib (no longer manufactured)
- Audio Excel DSP 16
- AudioDrive
- CMI8330 sound chip
- Compaq Deskpro XL onboard sound
- Corel Netwinder WaveArtist
- Crystal CS423x
- ESC614
- ESS1688 sound chip
- ESS1788 sound chip
- ESS1868 sound chip
- ESS1869 sound chip

- ESS1887 sound chip
- ESS1888 sound chip
- ESS688 sound chip
- ES1370 sound chip
- ES1371 sound chip
- Ensoniq AudioPCI (ES1370)
- Ensoniq AudioPCI 97 (ES1371)
- Ensoniq SoundScape (and compatibles made by Reveal and Spea)
- Gallant SC−6000
- Gallant SC−6600
- Gravis Ultrasound
- Gravis Ultrasound ACE
- Gravis Ultrasound Max
- Gravis Ultrasound with 16 bit sampling option
- HP Kayak
- Highscreen Sound−Booster 32 Wave 3D
- IBM MWAVE
- Logitech Sound Man 16
- Logitech SoundMan Games
- Logitech SoundMan Wave
- MAD16 Pro (OPTi 82C928, 82C929, 82C930, 82C924 chipsets)
- Media Vision Jazz16
- MediaTriX AudioTriX Pro
- Microsoft Windows Sound System (MSS/WSS)
- MiroSOUND PCM12
- Mozart (OAK OTI−601)
- OPTi 82C931
- Orchid SW32
- Personal Sound System (PSS)
- Pinnacle MultiSound
- Pro Audio Spectrum 16
- Pro Audio Studio 16
- Pro Sonic 16
- Roland MPU−401 MIDI interface
- S3 SonicVibes
- SY−1816
- Sound Blaster 1.0
- Sound Blaster 2.0
- Sound Blaster 16
- Sound Blaster 16ASP
- Sound Blaster 32
- Sound Blaster 64
- Sound Blaster AWE32
- Sound Blaster AWE64
- Sound Blaster PCI 128
- Sound Blaster Pro
- Sound Blaster Vibra16
- Sound Blaster Vibra16X
- TI TM4000M notebook
- Terratec Base 1
- Terratec Base 64

* ThunderBoard
* Turtle Beach Maui
* Turtle Beach MultiSound Classic
* Turtle Beach MultiSound Fiji
* Turtle Beach MultiSound Hurricane
* Turtle Beach MultiSound Monterey
* Turtle Beach MultiSound Pinnacle
* Turtle Beach MultiSound Tahiti
* Turtle Beach WaveFront Maui
* Turtle Beach WaveFront Tropez
* Turtle Beach WaveFront Tropez+
* VIA chip set
* VIDC 16−bit sound
* Yamaha OPL2 sound chip
* Yamaha OPL3 sound chip
* Yamaha OPL3−SA1 sound chip
* Yamaha OPL3−SA2 sound chip
* Yamaha OPL3−SA3 sound chip
* Yamaha OPL3−SAx sound chip
* Yamaha OPL4 sound chip

A word about compatibility: even though most sound cards are claimed to be *SoundBlaster compatible*, very few currently sold cards are compatible enough to work with the Linux SoundBlaster driver. These cards usually work better using the MSS/WSS or MAD16 driver. Only real SoundBlaster cards made by Creative Labs, which use Creative's custom chips (e.g. SoundBlaster16 Vibra), MV Jazz16 and ESS688/1688 based cards generally work with the SoundBlaster driver. Trying to use a SoundBlaster Pro compatible 16 bit sound card with the SoundBlaster driver is usually just a waste of time.

The Linux kernel supports the SCSI port provided on some sound cards (e.g. ProAudioSpectrum 16) and the proprietary interface for some CD−ROM drives (e.g. Soundblaster Pro). See the Linux SCSI HOWTO and CDROM HOWTO documents for more information.

A kernel driver to support joystick ports, including those provided on some sound cards, is included as part of the 2.2 kernels.

Note that the kernel SCSI, CD−ROM, joystick, and sound drivers are completely independent of each other.

3.2 Alternate Sound Drivers

Sound support in the Linux kernel was originally written by Hannu Savolainen. Hannu then went on to develop the Open Sound system, a commercial set of sound drivers sold by 4Front Technologies that is supported on a number of Unix systems. Red Hat Software sponsored Alan Cox to enhance the kernel sound drivers to make them fully modular. Various other people also contributed bug fixes and developed additional drivers for new sound cards. These modified drivers were shipped by Red Hat in their 5.0 through 5.2 releases. These changes have now been integrated into the standard kernel as of version 2.0. Alan Cox is now the maintainer of the standard kernel sound drivers, although Hannu still periodically contributes code taken from the commercial driver.

The commercial Open Sound System driver from 4Front Technologies tends to be easier to configure and support more sound cards, particularly the newer models. It is also compatible with applications written for the standard kernel sound drivers. The disadvantage is that you need to pay for it, and you do not get source code. You can download a free evaluation copy of the product before deciding whether to purchase it. For more information see the 4Front Technologies web page at http://www.opensound.com.

Jaroslav Kysela and others started writing an alternate sound driver for the Gravis UltraSound Card. The project was renamed *Advanced Linux Sound Architecture* (ALSA) and has resulted in what they believe is a more generally usable sound driver that can be used as a replacement for the built−in kernel drivers. The ALSA drivers support a number of popular sound cards, are full duplex, fully modularized, and compatible with the sound architecture in the kernel. The main web site of the ALSA project is http://www.alsa−project.org. A separate "Alsa−sound−mini−HOWTO" is available which deals with compiling and installing these drivers.

Markus Mummert (mum@mmk.e−technik.tu−muenchen.de) has written a driver package for the Turtle Beach MultiSound (classic), Tahiti, and Monterey sound cards. The documentation states:

> It is designed for high quality hard disk recording/playback without losing sync even on a busy system. Other features such as wave synthesis, MIDI and digital signal processor (DSP) cannot be used. Also, recording and playback at the same time is not possible. It currently replaces VoxWare and was tested on several kernel versions ranging from 1.0.9 to 1.2.1. Also, it is installable on UN*X SysV386R3.2 systems.

It can be found at http://www.cs.colorado.edu/~mccreary/tbeach.

Kim Burgaard (burgaard@daimi.aau.dk) has written a device driver and utilities for the Roland MPU−401 MIDI interface. The Linux software map entry gives this description:

> A device driver for true Roland MPU−401 compatible MIDI interfaces (including Roland SCC−1 and RAP−10/ATW−10). Comes with a useful collection of utilities including a Standard MIDI File player and recorder.

> Numerous improvements have been made since version 0.11a. Among other things, the driver now features IRQ sharing policy and complies with the new kernel module interface. Metronome functionality, possibility for synchronizing e.g. graphics on a per beat basis without losing precision, advanced replay/record/overdub interface and much, much more.

It can be found at ftp://metalab.unc.edu/pub/Linux/kernel/sound/mpu401−0.2.tar.gz.

Another novel use for a sound card under Linux is as a modem for amateur packet radio. The 2.1 and later kernels include a driver that works with SoundBlaster and Windows Sound System compatible sound cards to implement 1200 bps AFSK and 9600 bps FSK packet protocols. See the Linux AX25 HOWTO for details (I'm a ham myself, by the way −− callsign VE3ICH).

3.3 PC Speaker

An alternate sound driver is available that requires no additional sound hardware; it uses the internal PC speaker. It is mostly software compatible with the sound card driver, but, as might be expected, provides much lower quality output and has much more CPU overhead. The results seem to vary, being dependent on the characteristics of the individual loudspeaker. For more information, see the documentation provided with the release.

The software, which has not been updated for some time, can be found at ftp://ftp.informatik.hu−berlin.de/pub/Linux/hu−sound/.

3.4 Parallel Port

Another option is to build a digital to analog converter using a parallel printer port and some additional components. This provides better sound quality than the PC speaker but still has a lot of CPU overhead. The PC sound driver package mentioned above supports this, and includes instructions for building the necessary hardware.

4.Installation

Configuring Linux to support sound involves the following steps:

1. Installing the sound card.
2. Configuring Plug and Play (if applicable).
3. Configuring and building the kernel for sound support.
4. Creating the device files.
5. Booting the Linux kernel and testing the installation.

If you are running Red Hat Linux there is a utility called `sndconfig` that in most cases will detect your sound card and set up all of the necessary configuration files to load the appropriate sound drivers for your card. If you are running Red Hat I suggest you try using it. If it works for you then you can skip the rest of the instructions in this section.

If `sndconfig` fails, you are using another Linux distribution, or you want to follow the manual method in order to better understand what you are doing, then the next sections will cover each of these steps in detail.

4.1 Installing the Sound Card

Follow the manufacturer's instructions for installing the hardware or have your dealer perform the installation.

Older sound cards usually have switch or jumper settings for IRQ, DMA channel, etc; note down the values used. If you are unsure, use the factory defaults. Try to avoid conflicts with other devices (e.g. ethernet cards, SCSI host adaptors, serial and parallel ports) if possible.

Usually you should use the same I/O port, IRQ, and DMA settings that work under DOS. In some cases though (particularly with PnP cards) you may need to use different settings to get things to work under Linux. Some experimentation may be needed.

4.2 Configuring Plug and Play

Most sound cards now use the Plug and Play protocol to configure settings for i/o addresses, interrupts, and DMA channels. If you have one of the older sound cards that uses fixed settings or jumpers, then you can skip this section.

As of version 2.2 Linux does not yet have full Plug and Play support in the kernel. The preferred solution is to use the isapnp tools which ship with most Linux distributions (or you can download them from Red Hat's web site http://www.redhat.com/).

First check the documentation for your Linux distribution. It may already have Plug and Play support set up for you or it may work slightly differently than described here. If you need to configure it yourself,the details can be found in the man pages for the isapnp tools. Briefly the process you would normally follow is:

- Use `pnpdump` to capture the possible settings for all your Plug and Play devices, saving the result to the file /etc/isapnp.conf.
- Choose settings for the sound card that do not conflict with any other devices in your system and uncomment the appropriate lines in /etc/isapnp.conf. Don't forget to uncomment the `(ACT Y)` command near the end.
- Make sure that `isapnp` is run when your system boots up, normally done by one of the startup scripts. Reboot your system or run `isapnp` manually.

If for some reason you cannot or do not wish to use the isapnp tools, there are a couple of other options. If you use the card under Microsoft Windows 95 or 98, you can use the device manager to set up the card, then soft boot into Linux using the LOADLIN program. Make sure Windows and Linux use the same card setup parameters.

If you use the card under DOS, you can use the `icu` utility that comes with SoundBlaster16 PnP cards to configure it under DOS, then soft boot into Linux using the LOADLIN program. Again, make sure DOS and Linux use the same card setup parameters.

A few of the sound card drivers include the necessary software to initialize Plug and Play for the card. Check the documentation for that card's driver for details.

4.3 Configuring the Kernel

When initially installing Linux you likely used a precompiled kernel. These kernels often do not provide sound support. It is best to recompile the kernel yourself with the drivers you need. You may also want to recompile the kernel in order to upgrade to a newer version or to free up memory resources by minimizing the size of the kernel. Later, when your sound card is working, you may wish to rebuild the kernel sound drivers as modules.

The Linux Kernel HOWTO should be consulted for the details of building a kernel. I will just mention here some issues that are specific to sound cards.

If you have never configured the kernel for sound support before it is a good idea to read the relevant documentation included with the kernel sound drivers, particularly information specific to your card type. The files can be found in the kernel documentation directory, usually installed in /usr/src/linux/Documentation/sound. If this directory is missing you likely either have a very old kernel version or you have not installed the kernel source code.

Follow the usual procedure for building the kernel. There are currently three interfaces to the configuration process. A graphical user interface that runs under X11 can be invoked using `make xconfig`. A menu−based system that only requires text displays is available as `make menuconfig`. The original method, using `make config`, offers a simple text−based interface.

When configuring the kernel there will be many choices for selecting the type of sound card you have and the driver options to use. The on−line help within the configuration tool should provide an explanation of what each option is for. Select the appropriate options to the best of your knowledge.

After configuring the options you should compile and install the new kernel as per the Kernel HOWTO.

4.4 Creating the Device Files

For proper operation, device file entries must be created for the sound devices. These are normally created for you during installation of your Linux system. A quick check can be made using the command listed below. If the output is as shown (the date stamp will vary) then the device files are almost certainly okay.

```
% ls -l /dev/sndstat
crw-rw-rw-   1 root     root      14,   6 Apr 25  1995 /dev/sndstat
```

Note that having the right device files there doesn't guarantee anything on its own. The kernel driver must also be loaded or compiled in before the devices will work (more on that later).

In rare cases, if you believe the device files are wrong, you can recreate them. Most Linux distributions have a `/dev/MAKEDEV` script which can be used for this purpose.

4.5 Booting Linux and Testing the Installation

You should now be ready to boot the new kernel and test the sound drivers. Follow your usual procedure for installing and rebooting the new kernel (keep the old kernel around in case of problems, of course).

During booting, check for a message such as the following on powerup (if they scroll by too quickly to read, you may be able to retrieve them with the `dmesg` command):

```
Sound initialization started
<Sound Blaster 16 (4.13)> at 0x220 irq 5 dma 1,5
<Sound Blaster 16> at 0x330 irq 5 dma 0
<Yamaha OPL3 FM> at 0x388
Sound initialization complete
```

This should match your sound card type and jumper settings (if any).

Note that the above messages are not displayed when using loadable sound driver module (unless you enable it, e.g. using `insmod sound trace_init=1`).

When the sound driver is linked into the kernel, the `Sound initialization started` and `Sound initialization complete` messages should be displayed. If they are not printed, it means that there is no sound driver present in the kernel. In this case you should check that you actually installed the kernel you compiled when enabling the sound driver.

If nothing is printed between the `Sound initialization started` and the `Sound initialization complete` lines, it means that no sound devices were detected. Most probably it means that you don't have the correct driver enabled, the card is not supported, the I/O port is bad or that you have a PnP card that has not been configured.

The driver may also display some error messages and warnings during boot. Watch for these when booting the first time after configuring the sound driver.

Next you should check the device file /dev/sndstat. Reading the sound driver status device file should provide additional information on whether the sound card driver initialized properly. Sample output should look something like this:

```
% cat /dev/sndstat
Sound Driver:3.5.4-960630 (Sat Jan 4 23:56:57 EST 1997 root,
Linux fizzbin 2.0.27 #48 Thu Dec 5 18:24:45 EST 1996 i586)
Kernel: Linux fizzbin 2.0.27 #48 Thu Dec 5 18:24:45 EST 1996 i586
Config options: 0
```

```
Installed drivers:
Type 1: OPL-2/OPL-3 FM
Type 2: Sound Blaster
Type 7: SB MPU-401

Card config:
Sound Blaster at 0x220 irq 5 drq 1,5
SB MPU-401 at 0x330 irq 5 drq 0
OPL-2/OPL-3 FM at 0x388 drq 0

Audio devices:
0: Sound Blaster 16 (4.13)

Synth devices:
0: Yamaha OPL-3

Midi devices:
0: Sound Blaster 16

Timers:
0: System clock

Mixers:
0: Sound Blaster
```

The command above can report some error messages. "No such file or directory" indicates that you need to create the device files (see section 4.3). "No such device" means that sound driver is not loaded or linked into kernel. Go back to section 4.2 to correct this.

If lines in the "Card config:" section of `/dev/sndstat` are listed inside parentheses (such as "(SoundBlaster at 0x220 irq 5 drq 1,5)"), it means that this device was configured but not detected.

Now you should be ready to play a simple sound file. Get hold of a sound sample file, and send it to the sound device as a basic check of sound output, e.g.

```
% cat endoftheworld >/dev/dsp
% cat crash.au >/dev/audio
```

(Make sure you don't omit the ">" in the commands above).

Note that, in general, using `cat` is not the proper way to play audio files, it's just a quick check. You'll want to get a proper sound player program (described later) that will do a better job.

This command will work only if there is at least one device listed in the audio devices section of `/dev/sndstat`. If the audio devices section is empty you should check why the device was not detected.

If the above commands return "I/O error", you should look at the end of the kernel messages listed using the "dmesg" command. It's likely that an error message is printed there. Very often the message is "Sound: DMA (output) timed out – IRQ/DRQ config error?". The above message means that the driver didn't get the expected interrupt from the sound card. In most cases it means that the IRQ or the DMA channel configured to the driver doesn't work. The best way to get it working is to try with all possible DMAs and IRQs supported by the device.

Another possible reason is that the device is not compatible with the device the driver is configured for. This is almost certainly the case when a supposedly "SoundBlaster (Pro/16) compatible" sound card doesn't work with the SoundBlaster driver. In this case you should try to find out the device your sound card is compatible with (by posting to the comp.os.linux.hardware newsgroup, for example).

Some sample sound files can be obtained from
ftp://tsx−11.mit.edu/pub/linux/packages/sound/snd−data−0.1.tar.Z

Now you can verify sound recording. If you have sound input capability, you can do a quick test of this using commands such as the following:

```
# record 4 seconds of audio from microphone
EDT% dd bs=8k count=4 </dev/audio >sample.au
4+0 records in
4+0 records out
# play back sound
% cat sample.au >/dev/audio
```

Obviously for this to work you need a microphone connected to the sound card and you should speak into it. You may also need to obtain a mixer program to set the microphone as the input device and adjust the recording gain level.

If these tests pass, you can be reasonably confident that the sound D/A and A/D hardware and software are working. If you experience problems, refer to the next section of this document.

4.6 Troubleshooting

If you still encounter problems after following the instructions in the HOWTO, here are some things to check. The checks are listed in increasing order of complexity. If a check fails, solve the problem before moving to the next stage.

Step 1: Make sure you are really running the kernel you compiled.

You can check the date stamp on the kernel to see if you are running the one that you compiled with sound support. You can do this with the `uname` command:

```
% uname −a
Linux fizzbin 2.2.4 #1 Tue Mar 23 11:23:21 EST 1999 i586 unknown
```

or by displaying the file /proc/version:

```
% cat /proc/version
Linux version 2.2.4 (root@fizzbin) (gcc version 2.7.2.3) #1 Tue Mar 23 11:23:21 EST 1999
```

If the date stamp doesn't seem to match when you compiled the kernel, then you are running an old kernel. Did you really reboot? If you use LILO, did you re−install it (typically by running lilo)? If booting from floppy, did you create a new boot floppy and use it when booting?

Step 2: Make sure the kernel sound drivers are compiled in.

The easiest way to do this is to check the output of `dev/sndstat` as described earlier. If the output is not as expected then something went wrong with the kernel configuration or build. Start the installation process again, beginning with configuration and building of the kernel.

Step 3: Did the kernel detect your sound card during booting?

Make sure that the sound card was detected when the kernel booted. You should have seen a message on bootup. If the messages scrolled off the screen, you can usually recall them using the `dmesg` command:

```
% dmesg
```

or

```
% tail /var/log/messages
```

If your sound card was not found then something is wrong. Make sure it really is installed. If the sound card works under DOS then you can be reasonably confident that the hardware is working, so it is likely a problem with the kernel configuration. Either you configured your sound card as the wrong type or wrong parameters, or your sound card is not compatible with any of the Linux kernel sound card drivers.

One possibility is that your sound card is one of the *compatible* type that requires initialization by the DOS driver. Try booting DOS and loading the vendor supplied sound card driver. Then soft boot Linux using `Control−Alt−Delete`. Make sure that card I/O address, DMA, and IRQ settings for Linux are the same as used under DOS. Read the Readme.cards file from the sound driver source distribution for hints on configuring your card type.

If your sound card is not listed in this document, it is possible that the Linux drivers do not support it. You can check with some of the references listed at the end of this document for assistance.

Step 4: Can you read data from the dsp device?

Try reading from the /dev/audio device using the `dd` command listed earlier in this document. The command should run without errors.

If it doesn't work, then chances are that the problem is an IRQ or DMA conflict or some kind of hardware incompatibility (the device is not supported by Linux or the driver is configured for a wrong device).

A remote possibility is broken hardware. Try testing the sound card under DOS, if possible, to eliminate that as a possibility.

When All Else Fails

If you still have problems, here are some final suggestions for things to try:

- carefully re–read this HOWTO document
- read the references listed at the end of this document and the relevant kernel source documentation files
- post a question to one of the `comp.os.linux` or other Usenet newsgroups (comp.os.linux.hardware is a good choice; because of the high level of traffic in these groups it helps to put the string "sound" in the subject header for the article so the right experts will see it)
- Using a web/Usenet search engine with an intelligently selected search criteria can give very good results quickly. One such choice is http://www.altavista.digital.com
- try using the latest Linux kernel (but only as a last resort, the latest development kernels can be unstable)
- send mail to the author of the sound driver
- send mail to the author of the Sound HOWTO
- fire up emacs and type `Esc–x doctor` :–)

5.Applications Supporting Sound

I give here a sample of the types of applications that you likely want if you have a sound card under Linux. You can check the Linux Software Map, Internet archive sites, and/or files on your Linux CD–ROM for more up to date information.

As a minimum, you will likely want to obtain the following sound applications:

- audio file format conversion utility (e.g. `sox`)
- mixer utility (e.g. `aumix` or `xmix`)
- digitized file player/recorder (e.g. `play` or `wavplay`)
- MOD file player (e.g. `tracker`)

* MIDI file player (e.g. `playmidi`)

There are text−based as well as GUI−based versions of most of these tools. There are also some more esoteric applications (e.g. speech synthesis and recognition) that you may wish to try.

6.Answers To Frequently Asked Questions

This section answers some of the questions that have been commonly asked on the Usenet news groups and mailing lists.

Answers to more questions can also be found at the OSS sound driver web page.

6.1 What are the various sound device files?

These are the most standard device file names, some Linux distributions may use slightly different names.

/dev/audio

normally a link to /dev/audio0

/dev/audio0

Sun workstation compatible audio device (only a partial implementation, does not support Sun ioctl interface, just u−law encoding)

/dev/audio1

second audio device (if supported by sound card or if more than one sound card installed)

/dev/dsp

normally a link to /dev/dsp0

/dev/dsp0

first digital sampling device

/dev/dsp1

second digital sampling device

/dev/mixer

normally a link to /dev/mixer0

/dev/mixer0

first sound mixer

/dev/mixer1

second sound mixer

/dev/music

high−level sequencer interface

/dev/sequencer

low level MIDI, FM, and GUS access

/dev/sequencer2

normally a link to /dev/music

/dev/midi00

1st raw MIDI port

/dev/midi01

2nd raw MIDI port

/dev/midi02

3rd raw MIDI port

/dev/midi03

4th raw MIDI port

/dev/sndstat

displays sound driver status when read (also available as /proc/sound)

The PC speaker driver provides the following devices:

/dev/pcaudio

equivalent to /dev/audio

/dev/pcsp

equivalent to /dev/dsp

/dev/pcmixer

equivalent to /dev/mixer

6.2 How can I play a sound sample?

Sun workstation (.au) sound files can be played by sending them to the /dev/audio device. Raw samples can be sent to /dev/dsp. This will generally give poor results though, and using a program such as `play` is preferable, as it will recognize most file types and set the sound card to the correct sampling rate, etc.

Programs like wavplay or vplay (in the snd−util package) will give best results with WAV files. However they don't recognize Microsoft ADPCM compressed WAV files. Also older versions of play (from the Lsox package) doesn't work well with 16 bit WAV files.

The splay command included in the snd−util package can be used to play most sound files if proper parameters are entered manually in the command line.

6.3 How can I record a sample?

Reading /dev/audio or /dev/dsp will return sampled data that can be redirected to a file. A program such as `vrec` makes it easier to control the sampling rate, duration, etc. You may also need a mixer program to select the appropriate input device.

6.4 Can I have more than one sound card?

With the current sound driver it's possible to have several SoundBlaster, SoundBlaster/Pro, SoundBlaster16, MPU−401 or MSS cards at the same time on the system. Installing two SoundBlasters is possible but requires defining the macros SB2_BASE, SB2_IRQ, SB2_DMA and (in some cases) SB2_DMA2 by editing local.h manually. It's also possible to have a SoundBlaster at the same time as a PAS16.

With the 2.0 and newer kernels that configure sound using make config, instead of local.h, you need to edit the file /usr/include/linux/autoconf.h. After the section containing the lines:

```
#define SBC_BASE 0x220
#define SBC_IRQ (5)
#define SBC_DMA (1)
#define SB_DMA2 (5)
#define SB_MPU_BASE 0x0
#define SB_MPU_IRQ (-1)
```

add these lines (with values appropriate for your system):

```
#define SB2_BASE 0x330
#define SB2_IRQ (7)
#define SB2_DMA (2)
#define SB2_DMA2 (2)
```

The following drivers don't permit multiple instances:

- GUS (driver limitation)
- MAD16 (hardware limitation)
- AudioTrix Pro (hardware limitation)
- CS4232 (hardware limitation)

6.5 Error: No such file or directory for sound devices

You need to create the sound driver device files. See the section on creating device files. If you do have the device files, ensure that they have the correct major and minor device numbers (some older CD−ROM distributions of Linux may not create the correct device files during installation).

6.6 Error: No such device for sound devices

You have not booted with a kernel containing the sound driver or the I/O address configuration doesn't match your hardware. Check that you are running the newly compiled kernel and verify that the settings entered when configuring the sound driver match your hardware setup.

6.7 Error: No space left on device for sound devices

This can happen if you tried to record data to /dev/audio or /dev/dsp without creating the necessary device file. The sound device is now a regular file, and has filled up your disk partition. You need to run the script described in the *Creating the Device Files* section of this document.

This may also happen with Linux 2.0 and later if there is not enough free RAM on the system when the device is opened. The audio driver requires at least two pages (8k) of contiguous physical RAM for each DMA channel. This happens sometimes in machines with less than 16M of RAM or which have been running for very long time. It may be possible to free some RAM by compiling and running the following C program before trying to open the device again:

```
main() {
  int i;
  char mem[500000];
  for (i = 0; i < 500000; i++)
    mem[i] = 0;
  exit(0);
}
```

6.8 Error: Device busy for sound devices

Only one process can open a given sound device at one time. Most likely some other process is using the device in question. One way to determine this is to use the `fuser` command:

```
% fuser -v /dev/dsp
/dev/dsp:            USER       PID ACCESS COMMAND
                     tranter    265 f....  tracker
```

In the above example, the `fuser` command showed that process 265 had the device open. Waiting for the process to complete or killing it will allow the sound device to be accessed once again. You should run the `fuser` command as root in order to report usage by users other than yourself.

On some systems you may need to be root when running the `fuser` command in order to see the processes of other users.

6.9 I still get device busy errors!

According to Brian Gough, for the SoundBlaster cards which use DMA channel 1 there is a potential conflict with the QIC−02 tape driver, which also uses DMA 1, causing "device busy" errors. If you are using FTAPE, you may have this driver enabled. According to the FTAPE−HOWTO the QIC−02 driver is not essential for the use of FTAPE; only the QIC−117 driver is required. Reconfiguring the kernel to use QIC−117 but not QIC−02 allows FTAPE and the sound−driver to coexist.

6.10 Partial playback of digitized sound file

The symptom is usually that a sound sample plays for about a second and then stops completely or reports an error message about "missing IRQ" or "DMA timeout". Most likely you have incorrect IRQ or DMA channel settings. Verify that the kernel configuration matches the sound card jumper settings and that they do not conflict with some other card.

Another symptom is sound samples that *loop*. This is usually caused by an IRQ conflict.

6.11 There are pauses when playing MOD files

Playing MOD files requires considerable CPU power. You may have too many processes running or your computer may be too slow to play in real time. Your options are to:

- try playing with a lower sampling rate or in mono mode
- eliminate other processes
- buy a faster computer
- buy a more powerful sound card (e.g. Gravis UltraSound)

If you have a Gravis UltraSound card, you should use one of the mod file players written specifically for the GUS (e.g. `gmod`).

6.12 Compile errors when compiling sound applications

The version 1.0c and earlier sound driver used a different and incompatible `ioctl()` scheme. Obtain newer source code or make the necessary changes to adapt it to the new sound driver. See the sound driver Readme file for details.

Also ensure that you have used the latest version of soundcard.h and ultrasound.h when compiling the application. See the installation instructions at beginning of this text.

6.13 SEGV when running sound binaries that worked previously

This is probably the same problem described in the previous question.

6.14 What known bugs or limitations are there in the sound driver?

See the files included with the sound driver kernel source.

6.15 Where are the sound driver ioctls() etc. documented?

Currently the best documentation, other than the source code, is available at the 4Front Technologies web site, http://www.opensound.com. Another source of information is the *Linux Multimedia Guide*, described in the references section.

6.16 What CPU resources are needed to play or record without pauses?

There is no easy answer to this question, as it depends on:

- whether using PCM sampling or FM synthesis
- sampling rate and sample size
- which application is used to play or record
- Sound Card hardware
- disk I/O rate, CPU clock speed, cache size, etc.

In general, any 386 machine or better should be able to play samples or FM synthesized music on an 8 bit sound card with ease.

Playing MOD files, however, requires considerable CPU resources. Some experimental measurements have shown that playing at 44kHz requires more than 40% of the speed of a 486/50 and a 386/25 can hardly play faster than 22 kHz (these are with an 8 bit card sound such as a SoundBlaster). A card such as the Gravis UltraSound card performs more functions in hardware, and will require less CPU resources.

These statements assume the computer is not performing any other CPU intensive tasks.

Converting sound files or adding effects using a utility such as `sox` is also much faster if you have a math coprocessor (or CPU with on board FPU). The kernel driver itself does not do any floating point calculations, though.

6.17 Problems with a PAS16 and an Adaptec 1542 SCSI host adaptor

(the following explanation was supplied by `seeker@indirect.com`)

Linux only recognizes the 1542 at address 330 (default) or 334, and the PAS only allows the MPU−401 emulation at 330. Even when you disable the MPU−401 under software, something still wants to conflict with the 1542 if it's at its preferred default address. Moving the 1542 to 334 makes everyone happy.

Additionally, both the 1542 and the PAS−16 do 16−bit DMA, so if you sample at 16−bit 44 KHz stereo and save the file to a SCSI drive hung on the 1542, you're about to have trouble. The DMAs overlap and there isn't enough time for RAM refresh, so you get the dread ``PARITY ERROR − SYSTEM HALTED'' message, with no clue to what caused it. It's made worse because a few second−party vendors with QIC−117 tape drives recommend setting the bus on/off times such that the 1542 is on even longer than normal. Get the SCSISEL.EXE program from Adaptec's BBS or several places on the internet, and reduce the BUS ON time or increase the BUS OFF time until the problem goes away, then move it one notch or more further. SCSISEL changes the EEPROM settings, so it's more permanent than a patch to the DOS driver line in CONFIG.SYS, and will work if you boot right into Linux (unlike the DOS patch). Next problem solved.

Last problem − the older Symphony chipsets drastically reduced the timing of the I/O cycles to speed up bus accesses. None of various boards I've played with had *any* problem with the reduced timing except for the PAS−16. Media Vision's BBS has SYMPFIX.EXE that's supposed to cure the problem by twiddling a diagnostic bit in Symphony's bus controller, but it's not a hard guarantee. You may need to:

- get the motherboard distributor to replace the older version bus chip,
- replace the motherboard, or
- buy a different brand of sound card.

Young Microsystems will upgrade the boards they import for around $30 (US); other vendors may be similar if you can figure out who made or imported the motherboard (good luck). The problem is in ProAudio's bus interface chip as far as I'm concerned; *nobody* buys a $120 sound card and sticks it in a 6MHz AT. Most of them wind up in 25−40MHz 386/486 boxes, and should be able to handle *at least* 12MHz bus rates if the chips are designed right. Exit soapbox (stage left).

The first problem depends on the chipset used on your motherboard, what bus speed and other BIOS settings, and the phase of the moon. The second problem depends on your refresh option setting (hidden or synchronous), the 1542 DMA rate and (possibly) the bus I/O rate. The third can be determined by calling Media Vision and asking which flavor of Symphony chip is incompatible with their slow design. Be warned, though − 3 of 4 techs I talked to were brain damaged. I would be very leery of trusting *anything* they said about someone else's hardware, since they didn't even know their own very well.

6.18 Is it possible to read and write samples simultaneously?

The drivers for some sound cards support full duplex mode. Check the documentation available from 4Front Technologies for information on how to use it.

6.19 My SB16 is set to IRQ 2, but configure does not allow this value.

On '286 and later machines, the IRQ 2 interrupt is cascaded to the second interrupt controller. It is equivalent to IRQ 9.

6.20 If I run Linux, then boot DOS, I get errors and/or sound applications do not work properly.

This happens after a soft reboot to DOS. Sometimes the error message misleadingly refers to a bad CONFIG.SYS file.

Most of the current sound cards have software programmable IRQ and DMA settings. If you use different settings between Linux and MS−DOS/Windows, this may cause problems. Some sound cards don't accept new parameters without a complete reset (i.e. cycle the power or use the hardware reset button).

The quick solution to this problem it to perform a full reboot using the reset button or power cycle rather than a soft reboot (e.g. Ctrl−Alt−Del).

The correct solution is to ensure that you use the same IRQ and DMA settings with MS−DOS and Linux (or not to use DOS :−).

6.21 Problems running DOOM under Linux

Users of the port of ID software's game DOOM for Linux may be interested in these notes.

For correct sound output you need version 2.90 or later of the sound driver; it has support for the real−time `DOOM mode`.

The sound samples are 16−bit. If you have an 8−bit sound card you can still get sound to work using one of several programs available in ftp://metalab.unc.edu/pub/Linux/games/doom.

If performance of DOOM is poor on your system, disabling sound (by renaming the file `sndserver`) may improve it.

By default DOOM does not support music (as in the DOS version). The program `musserver` will add support for music to DOOM under Linux. It can be found at ftp://pandora.st.hmc.edu/pub/linux/musserver.tgz.

6.22 How can I reduce noise picked up by my sound card?

Using good quality shielded cables and trying the sound card in different slots may help reduce noise. If the sound card has a volume control, you can try different settings (maximum is probably best). Using a mixer program you can make sure that undesired inputs (e.g. microphone) are set to zero gain.

Philipp Braunbeck reported that on his ESS−1868 sound card there was a jumper to turn off the built−in amplifier which helped reduce noise when enabled.

On one 386 system I found that the kernel command line option `no−hlt` reduced the noise level. This tells the kernel not to use the halt instruction when running the idle process loop. You can try this manually when booting, or set it up using the command `append="no−hlt"` in your LILO configuration file.

Some sound cards are simply not designed with good shielding and grounding and are prone to noise pickup.

6.23 I can play sounds, but not record.

If you can play sound but not record, try these steps:

- use a mixer program to select the appropriate device (e.g. microphone)
- use the mixer to set the input gains to maximum
- If you can, try to test sound card recording under MS−DOS to determine if there is a hardware problem

Sometimes a different DMA channel is used for recording than for playback. In this case the most probable reason is that the recording DMA is set up incorrectly.

6.24 My "compatible" sound card only works if I first initialize under MS−DOS.

In most cases a "SoundBlaster compatible" card will work better under Linux if configured with a driver other than the SoundBlaster one. Most sound cards claim to be compatible (e.g. "16 bit SB Pro compatible" or "SB compatible 16 bit") but usually this SoundBlaster mode is just a hack provided for DOS games compatibility. Most cards have a 16 bit native mode which is likely to be supported by recent Linux versions (2.0.1 and later).

Only with some (usually rather old) cards is it necessary to try to get them to work in the SoundBlaster mode. The only newer cards that are the exception to this rule are the Mwave−based cards.

6.25 My 16−bit SoundBlaster "compatible" sound card only works in 8−bit mode under Linux.

16−bit sound cards described as SoundBlaster compatible are really only compatible with the 8−bit SoundBlaster Pro. They typically have a 16−bit mode which is not compatible with the SoundBlaster 16 and not compatible with the Linux sound driver.

You may be able to get the card to work in 16−bit mode by using the MAD16 or MSS/WSS driver.

6.26 Where can I find sound applications for Linux?

Here are some good archive sites to search for Linux specific sound applications:

- ftp://metalab.unc.edu/pub/Linux/kernel/sound/
- ftp://metalab.unc.edu/pub/Linux/apps/sound/
- ftp://tsx−11.mit.edu/pub/linux/packages/sound/
- ftp://nic.funet.fi/pub/Linux/util/sound/
- ftp://nic.funet.fi/pub/Linux/xtra/snd−kit/
- ftp://nic.funet.fi/pub/Linux/ALPHA/sound/

Also see the References section of this document.

6.27 Can the sound driver be compiled as a loadable module?

With recent kernels the sound driver is supported as several kernel loadable modules.

See the files in /usr/src/linux/Documentation/sound, especially the files Introduction and README.modules.

6.28 Can I use a sound card to replace the system console beep?

Try the `oplbeep` program, found at ftp://metalab.unc.edu/pub/Linux/apps/sound/oplbeep−2.3.tar.gz

Another variant is the `beep` program found at
ftp://metalab.unc.edu/pub/Linux/kernel/patches/misc/modreq_beep.tgz

The `modutils` package has an example program and kernel patch that supports calling an arbitrary external program to generate sounds when requested by the kernel.

Alternatively, with some sound cards you can connect the PC speaker output to the sound card so that all sounds come from the sound card speakers.

6.29 What is VoxWare?

The commercial version of the sound drivers sold by 4Front Technologies was previously known by other names such as *VoxWare*, *USS* (Unix Sound System), and even *TASD* (Temporarily Anonymous Sound Driver). It is now marketed as *OSS* (Open Sound System). The version included in the Linux kernel is sometimes referred to as *OSS/Free*.

For more information see the 4Front Technologies web page at http://www.opensound.com/. I wrote a review of OSS/Linux in the June 1997 issue of Linux Journal.

6.30 Sox/Play/Vplay reports "invalid block size 1024"

A change to the sound driver in version 1.3.67 broke some sound player programs which (incorrectly) checked that the result from the SNDCTL_DSP_GETBLKSIZE ioctl was greater than 4096. The latest sound driver versions have been fixed to avoid allocating fragments shorter than 4096 bytes which solves this problem with old utilities.

6.31 The mixer settings are reset whenever I load the sound driver module

You can build the sound driver as a loadable module and use `kerneld` to automatically load and unload it. This can present one problem – whenever the module is reloaded the mixer settings go back to their default values. For some sound cards this can be too loud (e.g. SoundBlaster16) or too quiet. Markus Gutschke (`gutschk@uni-muenster.de`) found this solution. Use a line in your /etc/conf.modules file such as the following:

```
options sound dma_buffsize=65536 && /usr/bin/setmixer igain 0 ogain 0 vol 75
```

This causes your mixer program (in this case `setmixer`) to be run immediately after the sound driver is loaded. The `dma_buffsize` parameter is just a dummy value needed because the option command requires a command line option. Change the line as needed to match your mixer program and gain settings.

If you have compiled the sound driver into your kernel and you want to set the mixer gains at boot time you can put a call to your mixer program in a system startup file such as /etc/rc.d/rc.local.

6.32 Only user root can record sound

By default the script in Readme.linux that creates the sound device files only allows the devices to be read by user `root`. This is to plug a potential security hole. In a networked environment, external users could conceivably log in remotely to a Linux PC with a sound card and microphone and eavesdrop. If you are not worried about this, you can change the permissions used in the script.

With the default setup, users can still play sound files. This is not a security risk but is a potential for nuisance.

6.33 Is the sound hardware on the IBM ThinkPad supported?

Information on how to use the mwave sound card on an IBM ThinkPad laptop computer can be found in the file /usr/src/linux/Documentation/sound/mwave, which is part of the kernel source distribution.

6.34 Applications fail because my sound card has no mixer

Some old 8−bit SoundBlaster cards have no mixer circuitry. Some sound applications insist on being able to open the mixer device, and fail with these cards. Jens Werner (werner@bert.emv.ing.tu−bs.de) reports a workaround for this is to link `/dev/mixer` to `/dev/null` and everything should work fine.

6.35 Problems with a SB16 CT4170

From Scott Manley (spm@star.arm.ac.uk):

> There seems to be a new type of Soundblaster − it was sold to us as a SB16 − the Model no. on the Card is CT4170. These Beasties only have one DMA channel so when you try to set them up then the kernel will have trouble accessing the 16 bit DMA. The solution is to set the second DMA to 1 so that the card will behave as advertised.

6.36 How to connect a MIDI keyboard to a soundcard

From Kim G. S. OEyhus (kim@pvv.ntnu.no):

> I looked all around the internet and in sound documentation on how to do something as simple as connecting the MIDI output from a master keyboard to the MIDI input on a sound card. I found nothing. The problem is that they both use the same device, /dev/midi, at least when using the OSS sound system. So I found a way to do it, which I want to share. This makes for a very simple synthesizer, with full MIDI support:
>
> CONNECTING A MIDI MASTER−KEYBOARD DIRECTLY TO A SOUNDCARD WITH MIDI
>
> A MIDI master−keyboard is a keyboard without any synthesizer, and with only a MIDI−out plug. This can be connected to the 15−pin D−SUB port on most sound−cards with a suitable cable.
>
> Such a keyboard can be used to control the MIDI synthesizer device for the card, thus making a simple keyboard controlled synthesizer.
>
> Compile the following program, say with 'gcc −o prog prog.c' and run it:

```
#include <fcntl.h>

main()
{
  int fil, a;
  char b[256];

  fil=open("/dev/midi", O_RDWR);
  for(;;)
    {
      a=read(fil, b, 256);
      write(fil, b, a);
    }
}
```

6.37 Problems with IRQ 15 and Ensoniq PCI 128

From Matthew Inger (mattinger@mindless.com):

> Information on getting an Ensoniq PCI 128 card to work.
>
> The problem that it was exhibiting was that it was trying to use interrupt 15 by default (Plug and Pray was responsible for this one). This interrupt is used by the secondary ide controller, and cannot be shared by other devices. You need to somehow force the es1370 to use another interrupt (should use interrupt 11 like it does under windows).
>
> I figured this one out for myself believe it or not.
>
> What I had to do was:
>
> a) in the BIOS, you have to tell the computer that you don't have a Plug and Play OS. I believe this is under advanced options in my BIOS.
>
> b) in the PCI settings in the BIOS, tell the computer to reserve interrupt 15 for legacy ISA devices. In my bios, under advanced options, there is a section for PCI settings. Under there, there is a Resource Exclusion area, and that's where to do this.
>
> When you reboot into linux you will be able to use sound. (I don't remember if it shows up in the boot messages or not like it used to). To be safe, I ran sndconfig again so that it would play the test message, which sounded not great, but it was there. When I played a CD however, it sounded perfect.
>
> Don't worry about windows, I tried both my cards: ISA Modem, and the Sound Card out, and they work without any hitches.
>
> The odds are your BIOS will be different from mine, but you just have to figure out where the settings are for the above two items. Good luck.

6.38 Where can I get freely available MIDI patches to run SoftOSS?

SoftOSS is a software−based wavetable synthesizer included with the kernel sound driver that is compatible with the Gravis Utrasound card. To operate the driver needs GUS compatible MIDI patch files. The documentation mentions the "public domain MIDIA patchset available from several ftp sites".

As explained on the 4Front Technologies web page http://www.opensound.com/softoss.html they can be downloaded from ftp://archive.cs.umbc.edu/pub/midia/instruments.tar.gz.

7.References

If you have a sound card that supports a CD−ROM or SCSI interface, the Linux SCSI HOWTO and the Linux CD−ROM HOWTO have additional information that may be useful to you.

The Sound Playing HOWTO describes how to play various types of sound and music files under Linux.

The Linux SoundBlaster AWE32/64 Mini−HOWTO describes how to get a SoundBlaster 32 or 64 card working under Linux.

Programming information is available from the 4Front Technologies web site at http://www.opensound.com/pguide.

The following FAQs are regularly posted to the Usenet newsgroup news.announce as well as being archived at ftp://rtfm.mit.edu/pub/usenet/news.answers:

- PCsoundcards/generic−faq (Generic PC Soundcard FAQ)
- PCsoundcards/soundcard−faq (comp.sys.ibm.pc.soundcard FAQ)
- PCsoundcards/gravis−ultrasound/faq (Gravis UltraSound FAQ)
- audio−fmts/part1 (Audio file format descriptions)
- audio−fmts/part2 (Audio file format descriptions)

The FAQs also list several product specific mailing lists and archive sites. The following Usenet news groups discuss sound and/or music related issues:

- alt.binaries.sounds.* (various groups for posting sound files)
- alt.binaries.multimedia (for posting Multimedia files)
- alt.sb.programmer (Soundblaster programming topics)
- comp.multimedia (Multimedia topics)
- comp.music (Computer music theory and research)
- comp.sys.ibm.pc.soundcard.* (various IBM PC sound card groups)

A web site dedicated to multimedia can be found at http://viswiz.gmd.de/MultimediaInfo/. Another good site for Linux MIDI and sound applications is http://sound.condorow.net/. Creative Labs has a web site at http://www.creaf.com/. MediaTrix has a web site at http://www.mediatrix.com/.

The Linux mailing list has a number of "channels" dedicated to different topics, including sound. To find out how to join, send a mail message with the word "help" as the message body to majordomo@vger.rutgers.edu. These mailing lists are not recommended for questions on sound card setup etc., they are intended for development related discussion.

As mentioned several times before, the kernel sound driver includes a number of Readme files containing useful information about the sound card driver. These can typically be found in the directory /usr/src/linux/drivers/sound.

Information on OSS, the commercial sound driver for Linux and other Unix compatible operating systems, can be found at the 4Front Technologies web page at http://www.opensound.com/.

The *Linux Software Map* (LSM) is an invaluable reference for locating Linux software. Searching the LSM for keywords such as *sound* is a good way to identify applications related to sound hardware. The LSM can be found on various anonymous FTP sites, including ftp://metalab.unc.edu/pub/Linux/docs/LSM/ (formerly known as sunsite). There are also various web sites that maintain databases of Linux applications. One such site is http://www.freshmeat.net.

The Linux Documentation Project has produced several books on Linux, including *Linux Installation and Getting Started*. These are freely available by anonymous FTP from major Linux archive sites or can be purchased in hardcopy format.

Finally, a shameless plug: If you want to learn a lot more about multimedia under Linux (especially CD−ROM and sound card applications and programming), check out my book *Linux Multimedia Guide*, ISBN 1−56592−219−0, published by O'Reilly and Associates. As well as the original English version, French and Japanese translations are now in print. For details, call 800−998−9938 in North America or check the web page http://www.ora.com/catalog/multilinux/noframes.html or my home page http://www.pobox.com/~tranter.