Dave Phillips
400 Glessner Avenue
Findlay, OH 45840 USA
dlphilp@bright.net

Computer Music and the Linux Operating System: A Report from the Front

The purpose of this article is to introduce readers to Linux and to describe the factors that make this operating system attractive to computer musicians. I have divided this exposition into two main parts. The first describes and assesses Linux as a general platform for computing. The second, longer section focuses on Linux's audio and music capabilities. In that section, a historical overview precedes a necessarily brief account of the Linux audio software base, followed by a prediction for the near future of Linux audio development and the acceptance of Linux for music applications.

A General Introduction to Linux

Many readers of this journal will need no introduction to Linux, but for the rest a brief description is in order. I will first review Linux in general terms, outlining its advantages as well as aspects that historically have been of concern. The remainder of this article will then describe in detail the considerations specific to audio and music.

Linux is a UNIX-like operating system designed to provide computer users with a free system comparable to traditional (and usually expensive) UNIX systems such as IBM's AIX, Sun's Solaris, or SGI's IRIX. ''Free'' here means that the system software itself is freely available at no cost, with free access to the source code, and with the assurance that anyone's development of the system benefits everyone else who develops or uses the system. Linux is compliant with the Portable Operating System Interface (POSIX), a standard of the Institute of Electrical and Electronics Engineers (IEEE).

The system kernel was developed in the early 1990s by Linus Torvalds at the University of Helsinki in Finland, and the system was essentially completed by making use of components developed for GNU (a recursive acronym for GNU's Not UNIX), which itself is a project of Richard Stallman's Free Software Foundation. Other notable contributions toward a complete system include the X Windows System (a client-server windowing engine) from the XFree86 organization, and networking tools and utilities from Berkeley Standard Distribution (BSD) UNIX. Originally designed for Intel's i386 processors, Linux now runs on many other CPUs and platforms, including the PowerPC, 64-bit Alpha systems, IBM's S/390, various embedded systems, and even hand-held machines (palmtops).

It should be noted that many users (and some distributions) refer to the platform as ''GNU/Linux'' in recognition of the Free Software Foundation's significant contribution to the system. Strictly speaking, Linux is only the kernel, whereas most of its surrounding system software has indeed come from the GNU project's efforts.

Although the Linux kernel source code is in fact available at no cost, most new users begin their Linux experience by purchasing a software-plus-manual-plus-support bundle called a Linux distribution. A distribution is normally a complete system including the X11 graphics and windowing system, a robust TCP/IP networking subsystem, and broad support for hardware extensions. Distributions also typically include a wealth of applications, utilities, and programming tools far in excess of comparable commercial systems. Linux itself is legally protected by the remarkable ''copyleft'' provisions of the GNU Public License (GPL), but available applications for Linux may be covered by any one of a variety of licenses ranging from freeware with absolutely no restrictions to commercial offerings protected by the end-user license agreements (EULA) commonly associated with off-the-shelf software. The great significance of the Linux/GPL marriage lies in the stipulation that any source-level modifications to the system must in turn be freely available upon redistribution. By establishing a source-code base that is publicly open and extensible by contributors, Linux has developed rapidly since its initial public release. Distributions are now available commercially from a number of

Computer Music Journal, 27:4, pp. 27–42, Winter 2003

high-profile vendors such as Red Hat, Mandrake, and SuSE, and extensive documentation about Linux is available online and in print.

The Linux applications software base has benefited from a rich inheritance of tools and utilities designed for UNIX computing environments, including sophisticated packages such as the Emacs text editor, the X11 windowing system, and the GNU C compiler (gcc). Developers have transformed the Linux user's environment from one dominated by an image of the daunting UNIX command prompt into one now media-rich and apparently infinitely customizable. The Linux graphical user interface (GUI) can be tailored to available resources and needs; it may consist of a simple window manager (e.g., BlackBox, FVWM, or IceWM) or a complete desktop environment such as the K Desktop Environment (KDE) or the GNU Network Object Model Environment (GNOME). Of course, users are free to dispense with a GUI completely and run the system entirely from a text-mode console.

Linux development has naturally strived toward interoperability, supporting a wide variety of file systems (Windows VFAT, MS-DOS, the Reiser journaling filesystem, etc.) and network protocols (TCP/IP, UDP, Samba, etc.). The system integrates well with other platforms, promoting the use of each platform for its particular strengths.

Scalability

In 1992, Linus Torvalds stated that his evolving brainchild would likely continue to run only on Intel i386-based hardware. But by mid 2003, the system has been deployed on a surprising number of platforms. Linux still runs on x86 machines, but now PowerPCs, Alpha hardware, Macintosh computers, and even the Amiga enjoy the benefits of the system. Ports to unusual hardware are often undertaken as interesting programming projects, and some ports have been written for every type of hardware from palm-tops to supercomputers.

Linux scales well in almost any computing environment, from the single-user desktop to huge clusters and processor "farms." The system has already scored major successes in the professional 3D-rendering industry, with large Linux-based rendering farms proving to be cheaper, faster, and more reliable than any other solution. In-house programmers for post-production facilities have found the Linux source code to be a particular blessing: they are no longer bound to wait for fixes and enhancements to arrive at the convenience of a commercial provider. And thanks to the GPL, they are free to make whatever changes are needed to achieve the required performance.

Manufacturers requiring an embedded operating system have also embraced Linux. The availability of source code enables easier customization, and the GPL relieves companies from burdensome and restrictive licenses. NEC, Sony, Panasonic, and IBM have already employed embedded Linux in devices such as mobile phones and network routers.

Security

Computer security is a primary concern to system administrators of environments networked to the Internet or on local intranets. Once again, the open-source nature of Linux proves most amenable to resolving this concern: the code base is continually pored over by the many eyes of its development community and tested by thousands of users in demanding real-world situations, including high-security corporate and military installations. Again, system administrators are freed from depending on a vendor's schedule for security updates and patches, though it must be stressed that no system (including Linux) is automatically secure. However, with the proper precautions, Linux can be easily configured as a safe, solid, and famously reliable network server and gateway.

Cost of Ownership

The total cost of ownership (TCO) of Linux has been variously reported as both higher and lower than that of other systems, particularly Windows. How the relevant figures are computed indicates the biases necessarily inherent to such reports, but by and large they all point to a similar conclusion: if an organization already includes Linux-savvy personnel, its startup and maintenance costs will be low. However, training a staff that is totally unfamiliar with the system is likely to result in a larger initial outlay of time and energy. Still, such an expenditure may be desirable to enjoy the benefits of an open-source solution and to avoid the increasingly onerous costs and licensing terms from single-source vendors. Finally, it is indeed true that Linux itself is cost-free. Users can download the kernel source code for free, a single disc may be freely redistributed (i.e., with no per-seat licensing), and many distributors even place their commercially available products online as free downloads. The high quality of the system and the scale of its continual development, combined with this aspect of the Linux TCO, creates an increasingly difficult and competitive field for commercial vendors of closed-source operating systems.

Problems and Concerns

In 1995, installing Linux was something of a labor of love. Disk partitioning, file system definition, manual configuration of sound and video cards, and other preparatory processes guaranteed that only the intrepid dared to try it. Fortunately, today's installation procedures are far simpler. Hardware is now automatically detected, and system configuration is often automatic, making the process as near painless as possible for the beginner, yet the system remains fully customizable by the more experienced user. Contemporary distributions such as Mandrake and Red Hat are quite easy to install, and the GNOME and KDE desktop environments are now typically the new user's first experience in Linux. The familiar ''point-and-click'' GUI creates a comfortable learning area for users migrating from other GUI-based systems such as Mac OS or Microsoft Windows.

Some vendors even sell machines pre-loaded with a Linux distribution, obviating the installation phase; however, most users are likely to go through the process themselves on their own machines. By comparison, relatively few users of Windows or the Macintosh ever install those systems themselves, so ease of installation has been a primary goal for distribution vendors.

Technical support has also been something of a thorny issue for Linux. Without a central location or single company in charge of Linux, there is apparently no one to call when things do not work as they ought. However, most mainstream distributions now include limited support for the typical desktop user, with full support contracts available to enterprise users. Some companies offer Linux technical support as their primary business model, but for many users, the most valuable support comes from the helpful resources on the Internet. Linux-specific newsgroups, mail lists, and Internet Relay Chat (IRC) channels are available for every level of user and developer, from ''newbie'' to ''wizard.'' The popular Google Internet search engine (available online at www.google.com) provides excellent Linux-specific searches throughout the Web and Google's own massive archive of newsgroup traffic.

In its earlier years, Linux was justly criticized for its rather difficult learning path, but that criticism is largely no longer justified. An enormous amount of material, online and in print, is now available to new users and developers. Monthly magazines dedicated to Linux are available, and new Linux-specific books appear regularly on major booksellers's shelves. Certification of Linux technical proficiency is available from companies such as Sair and Red Hat, guaranteeing competence at various levels of deployment.

Support from hardware manufacturers remains a problematic issue. The classic ''chicken-and-egg'' syndrome is hard at work here, notably with regard to audio equipment. Manufacturers commonly write and distribute drivers for Windows and the Macintosh but not for Linux. Depending on the willingness of the manufacturer, one of three typical scenarios results: the manufacturer decides to write and distribute Linux drivers; the manufacturer hands the needed specifications to Linux developers for them to develop the driver; or the manufacturer has no interest in either of the other scenarios and provides no support for Linux. Unfortunately, this last scenario has been most common with regard to audio and MIDI hardware, but growing media coverage of the system may widen its user base, sufficiently increasing demand to the point of recognition by major audio hardware manufacturers. Given the rate and quality of the system's development as an audio platform, it is certainly becoming more difficult to ignore.

Linux as a Platform for Audio and Music

Now that we have reviewed Linux as a general computing platform, we will examine its suitability for computer music in particular. This section traces the evolution of Linux's audio capabilities and describes the spectrum of available software, from low-level tools through commercial music applications.

General Factors Favoring Linux for Music

Availability of an operating system's source code is not usually regarded as an important factor to aspiring computer musicians, but it has proved critical to developers working on low-latency performance enhancements at the Linux kernel level. Developers interested in improving Linux multimedia performance have been very active modifying the kernel sources, a process utterly impossible to implement in closed-source commercial operating systems (i.e., impossible without paying high fees) such as Windows or the Macintosh OS. Given this free and open-source availability of code, it is theoretically possible for an ambitious computer music student to design a modified version of Linux optimized to do nothing but process audio and MIDI.

Other aspects of Linux's UNIX heritage can be found in its support for various mechanisms supporting inter-process communication. Pipes and other redirectors can be used at the command line to create complex processing chains; for example, Csound supports line-events, a mechanism for feeding score events to the program directly from the command prompt in real time, dispensing with the traditional Csound score. Reliable piping and redirection is characteristic of Linux and other UNIX systems.

History of Linux Audio

Thanks to the initial labors of Hannu Savolainen in 1992, the Linux audio applications programming interface (API) provided support for the basic SoundBlaster-compatible sound devices. At that time, those devices included a pulse-code modulation (PCM) recording and playback device (/dev/dsp), a raw MIDI input/output (I/O) device (/dev/midi), and an audio device mixer (/dev/mixer). This kernel audio API, known as the Open Sound System interface (OSS/Free), has been included with the kernel sources since 1992. However, in 2002, Linus Torvalds announced the inclusion of the Advanced Linux Sound Architecture (ALSA) API into the 2.5.x development series of the Linux kernel, setting the stage for the adoption of the ALSA API and drivers into the stable kernel releases (2.6.x and onwards). (See www.alsa-project.org for information on ALSA.) This event is particularly important to Linux audio development: it signifies the arrival of a freely available, open-source API for sound and MIDI programming in Linux with fully professional-level capabilities.

Note: the Linux kernel is available in an even-numbered stable series (e.g., 2.4.x) and an odd-numbered development track (e.g., 2.5.x). Separating the stable and development series in this way ensures reliability for the normal user without the risks associated with incomplete or experimental features under development and testing.

Hannu Savolainen and other Linux kernel hackers had expanded the capabilities of the OSS/Free API primarily to accommodate features found on popular sound cards. Thanks to the willingness of some companies to donate the necessary specifications, OSS/Free could provide support for some relatively advanced features, such as on-board synthesizers and full-duplex recording. Unfortunately, most manufacturers of professional digital audio hardware refused to provide free access to relevant specs and documentation, resulting in a classic loop: Linux applications developers could not write high-performance sound software without support for more powerful hardware, whereas the hardware manufacturers cited the absence of high-performance software as evidence for insufficient demand. One way out of this impasse was taken by Hannu Savolainen: he formed a company that would sign non-disclosure agreements to provide Linux audio developers and users with a greater selection of sound drivers, albeit in the form of commercial software with some binary-only closed-source components. This package, called OSS/Linux, has itself evolved from the OSS/Free API and now supports a number of professional audio boards and chipsets.

ALSA began as Jaroslav Kysela's project to provide better support for his Gravis Ultrasound sound card. Eventually, he found himself at the center of a team of talented developers dedicated to the goal of a completely free open-source sound driver and API that would provide professional-grade services to Linux audio programmers and users. Thanks to the efforts of Mr. Kysela's team and the foresight of a few enlightened manufacturers, the project is well on its way to meeting and surpassing its original goals. Most popular sound-card chipsets are already supported, and I have already mentioned the adoption of ALSA into the kernel sources. Other significant ALSA achievements include the drivers for the M-Audio Delta series of multichannel audio boards, Takashi Iwai's support for the Creative Labs SBLive and Audigy sound cards, and Paul Davis's drivers for the RME Hammerfall series of professional digital audio boards. Documentation to assist in getting started with ALSA is available online at www.suse.de/~mana/alsa090_howto.html.

The Modern Linux Sound System

Linux audio developers have been evolving a professionally capable sound system based entirely on free software. The system's current status presents a layered approach to solving the problems inherent to the increased system performance demands made by audio and video applications. Thus we can view the modern Linux sound system at three levels: the kernel level (e.g., basic system hooks and driver modules); a middle layer (e.g., ALSA, JACK, LADSPA, libsndfile, esd, and artsd); and user space (programs such as Ardour, Sweep, and MusE).

Development on each of these levels has been deep and extensive. The rewards of ALSA development at the kernel level have already been mentioned. ALSA also has a significant presence in the middle layer with its exceptional utilities and tools such as aconnect (a mechanism for routing MIDI I/O between ALSA-aware programs) and alsamixer (a text-mode system audio mixer). Other notable middle-level software includes the JACK Audio Connection Kit (JACK), sound servers such as KDE's Analog Real-Time Synthesizer daemon (aRts) and GNOME's Enlightened Sound Daemon (esd), libraries such as libsndfile and libaudiofile, and plug-ins based on the Linux Audio Developer's Simple Plug-in API (LADSPA). (See www.ladspa.org for more information on LADSPA.) Note that there is no explicitly prescribed middle level, but I find it convenient to refer to it when describing essential software that is itself neither kernel-space property nor a direct concern of the normal user.

Paul Davis's JACK, available from jackit.sourceforge.net, deserves special description. This software provides a mechanism for connecting the audio I/O of applications without affecting performance latency. For example, it is possible to route the audio output from a MIDI sequencer running a software synthesizer to the recording channel input of a hard-disk recorder that itself may be recording an audio stream from yet another concurrently running application. JACK ensures low-latency and sample-synchronous execution for all streams without loss of data, and it has been designed to provide a method of transport control between JACK-aware applications. Audio multiplexing is also characteristic of the aRts and esd system sound servers, but although those solutions are suitable for common desktop audio, they are not robust enough to handle the demands of professional audio applications (particularly hard-disk recording systems).

Erik de Castro Lopo's libsndfile is representative of essential middle layer audio software. A collection of such software has been growing steadily, to

Table 1. Some Web resources for Linux audio

4Front Technologies	www.opensound.com
"Achieving Low-Latency Response Times Under Linux"	linux.oreillynet.com/pub/a/linux/2000/11/17/low_latency.html
A GNU/Linux Audio Distribution (AGNULA)	www.agnula.org
The Advanced Linux Sound Architecture (ALSA) Project	www.alsa-project.org
"ALSA 0.9.0 Programming HOWTO"	www.suse.de/~mana/alsa090_howto.html
Debian Multimedia Distribution (Demudi)	www.demudi.org
"Introduction to MIDI Programming in Linux"	ccrma-ww.stanford.edu/~craig/articles/linuxmidi
JACK Audio Connection Kit	jackit.sourceforge.net
Linux Audio Developer's Simple Plug-in API (LADSPA)	www.ladspa.org
The Linux Audio Developers Mailing List	www.linuxdj.com/audio/lad
"The Linux Audio Quality HOWTO"	www.linuxdj.com/audio/quality
"The Linux Audio Users Guide"	www.boosthardware.com/LAU/Linux_Audio_Users_Guide
The Linux Audio Users Mailing List	www.linuxdj.com/audio/lad/user.php3
Planet CCRMA	www-ccrma.stanford.edu/planetccrma/software
Turn-Key Linux Audio	lulu.esm.rochester.edu/kevine/turnkey

the point where Linux audio developers now have excellent open-source libraries for sound file I/O (libsndfile, libaudiofile), sample rate conversion (libsamplerate), and time/pitch stretching (the SoundTouch library). This type of software works in both directions, aiding developers by providing commonly available high-quality audio libraries and giving users a set of libraries shared by various applications.

In 1996, I began listing the available user-space audio applications for Linux. At that time, the list totaled fewer than 50 items, ranging from completely developed programs such as UNIX audio stalwarts MiXViews and Csound to barely usable prototypes and alpha releases. In mid 2003, the list includes more than 1,000 hyperlinks to a wide variety of software that still ranges from the newly born to the fully developed. This situation is not very different from the Windows and Macintosh worlds with their various sites and repositories for freeware and shareware applications, but there is one tremendous difference: nearly all Linux audio and MIDI software includes the source code, making it possible for any knowledgeable user to make any necessary or desired changes to the software at any time.

The maturity of Linux sound applications has itself evolved along with the various Linux development toolkits. In particular, GUI toolkits such as GTK, FLTK, and QT now provide attractive interface components that lend a more sophisticated appearance to applications. Multimedia toolkits such as SDL (Simple DirectMedia Layer), CSL (Common Sound Layer), and OpenAL have also brought more capabilities to audio and video performance under Linux while providing programmers with tools for simpler access to those capabilities.

The variety of user-level applications has also grown steadily since 1995. Nearly all the gaps in the early lists have been filled, and Linux audio and MIDI software users now enjoy full-featured, professional-grade hard-disk recording , sound-file editors, MIDI sequencers, software sound synthesizers, effects plug-ins, network broadcasting, and audio software in many other categories, far too many to enumerate here. A subsequent section of this article details some specific applications from the categories mentioned, and I urge the interested reader to consult the latest edition of the Linux Sound and Music Applications Web site (linux-sound.org) for a complete overview of the categories and varieties of Linux audio software. A sampling of resources on the Web for Linux and audio is provided in Table 1. Readers might also be interested in my publication *The Book of Linux Music and Sound*, published in 2000 by No Starch Press in San Francisco.

Emulation of Other Operating Environments

Some excellent operating system emulators are available for Linux, but the performance of multimedia applications under an emulated operating system remains problematic. Some MS-DOS MIDI and sound applications will run very well under the DOSemu emulator, and some non-real-time audio applications for the Macintosh will run well under the ARDI Executor Mac emulation environment, but applications such as Cubase and Cakewalk are still beyond the capabilities of the available Windows emulators. However, using program code from the Windows Emulator (WINE), the CodeWeavers development team has created software (the Crossover Plug-in) that successfully invokes the QuickTime 5 player and the Windows Media Player under Linux. The VMWare program provides yet another level of interoperability: this software opens a virtual machine that can run Windows applications as though they were opened in their native environment. Again, performance of real-time audio and visual applications is compromised to the extent of being unusable in anything like a professional or semi-professional studio. The STEEM and UAE programs emulate the Atari ST and Amiga computers, and many MIDI and sound applications for those machines will run quite well in the emulated environments.

Audio Latency

The Linux kernel can be patched to reduce audio latency to professionally acceptable levels, down to 3 msec or less. The kernel provides access to a real-time clock for fine-grained timing for sequencers and other real-time applications, and hard-disk tuning utilities can dramatically optimize disk performance. Combining these factors with the system's exceptional stability, Linux becomes a natural choice for the design of a low-cost high-performance digital audio workstation (DAW).

An un-patched, un-tuned Linux 2.4 kernel can create latencies of up to 300 msec, far beyond any acceptable limit for professional audio applications. Fortunately, patches from Ingo Molnar and Andrew Morton (see linux.oreillynet.com/pub/a/linux/2000/11/17/low_latency.html) have proven that the Linux kernel can be tuned to eliminate performance bottlenecks, particularly when scheduled processes unnecessarily remain active, blocking access to other system activity and creating high-latency conditions. These low-latency patches are now available for anyone to apply, and they are normally coupled with the use of the hdparm utility for fine-tuning disk performance. The hdparm program gives users control over data transfer rates and sizes, allowing drives to be tuned for maximum performance with maximum security.

Other research has also brought other solutions to the latency problem. One such solution is Victor Yodaiken's RTLinux, a Linux system optimized for real-time work where latencies must be measured in microseconds, not milliseconds. Unfortunately, RTLinux is not an option for Linux audio developers, because sound card drivers must be rewritten to take advantage of RTLinux timing, and the gains are not generally regarded as worth the effort. The MontaVista Linux distribution (www.mvista.com) is another example of a version of Linux highly optimized for real-time performance, though audio latency is not a direct concern of the developers of MontaVista. The performance of MontaVista Linux certainly impresses the Yamaha Corporation: MontaVista has been chosen to be the embedded operating system for a wide variety of future Yamaha products. It may not be too unbelievable that MontaVista could show up in some of Yamaha's next-generation line of digital synthesizers.

Robert Love has introduced a set of patches for giving certain processes preemptive status at the kernel level. A popular approach to optimal latency reduction is to apply these patches along with Andrew Morton's low-latency set. This combination has been shown to maintain very low-latency over very long periods of time. In fact, the 2.5.x development kernel series has introduced Robert Love's preemptive patch as a kernel configuration option. In addition, the 2.5.x series officially dispenses with the aging OSS/Free API and introduces ALSA as the new kernel sound system. The real-time clock (RTC) is still a configuration option, and a finer-grained POSIX clock has also been added to the most recent development releases.

Audio Drivers and Libraries

To reiterate, the three primary sources for sound card and audio board drivers are the OSS/Free kernel modules, the ALSA drivers and library, and the OSS/Linux commercial driver package. Each source has its strengths and weaknesses, but ALSA is unquestionably becoming the standard set of audio and MIDI drivers for Linux. Stand-out features of ALSA include support for audio interfaces from consumer-grade sound cards to professional digital audio boards; multi-processor, thread-safe capability; fully modularized drivers; a user-space library to simplify audio applications programming and provide high-level functionality; and compatibility with the older OSS/Free API. Additionally, ALSA supports serial port and Universal Serial Bus (USB) interfaces, and it is completely free and open-source, with its code base licensed under the GPL or the GNU Lesser General Public License (LGPL).

The OSS/Free API should now be regarded as obsolete, and authors of new Linux audio software are well advised to learn and utilize the ALSA API. OSS/Linux is a fine package, and it may be the only solution where specific hardware may be unsupported by either OSS/Free or ALSA, but it is definitely not an open-source solution. A few other independent driver projects exist, but they are usually for specific cards and do not propose a generalized API.

Model Linux Audio Setups

ALSA support for professional audio hardware deserves further comment. Multichannel boards such as the RME Hammerfall and the M-Audio Delta series have excellent ALSA drivers, so it is now possible to work with professional-grade hardware under Linux. I shall describe some of the software that works with those boards, but first I want to describe two typical Linux studio systems, one for advanced desktop work and another for professional duty.

Both systems should include a fast computer, large and fast hard-disk(s), at least 256 MB of random-access memory (RAM), a good video card and monitor (preferably with support for accelerated OpenGL), and optionally a MIDI keyboard. Basic software should include a recent Linux kernel (2.4.x series and up) with real-time clock support and patched for low-latency; XFree86 4.0 or higher; ALSA (0.9.x or higher) drivers, library, and utilities; JACK; and the LADSPA plug-ins.

Beyond these specifications, the main difference between the advanced desktop system and the professional system will be in the choice of sound card or audio board. Some consumer-grade sound cards can be quite effective even for semi-professional work. Signal-to-noise ratios can be bearable, and S/PDIF digital audio interfaces show up on a few cards (e.g., Creative's SBLive! and Audigy), making possible a digitally-connected Linux home studio. However, for work at professional standards, other digital audio boards are necessary (e.g., the RME Hammerfall or the M-Audio Delta cards). A professional-level studio is also likely to include considerably more expensive and powerful outboard equipment not within the scope of this article.

The question often arises as to which Linux distribution is best suited for audio work. The truth is that any of the available distributions can be customized to the extent needed by the user. Happily, the trend is to supply the user with complete turnkey solutions, relieving the user from the burden of searching for and installing the necessary components for a totally Linux-based sound system. The Debian Multimedia Distribution (Demudi) project led the way (see www.demudi.org), and there are now at least three other turnkey solutions available for users.

Noncommercial Audio and Music Applications

Given the large number of Linux audio applications, it is not possible for this article to give more than a glimpse of the variety of such software available. I refer the interested reader to the Linux Sound and Music Applications Web site for a better overview of the available software. However, it is worthwhile to consider some of the larger categories and take a peek at their contents.

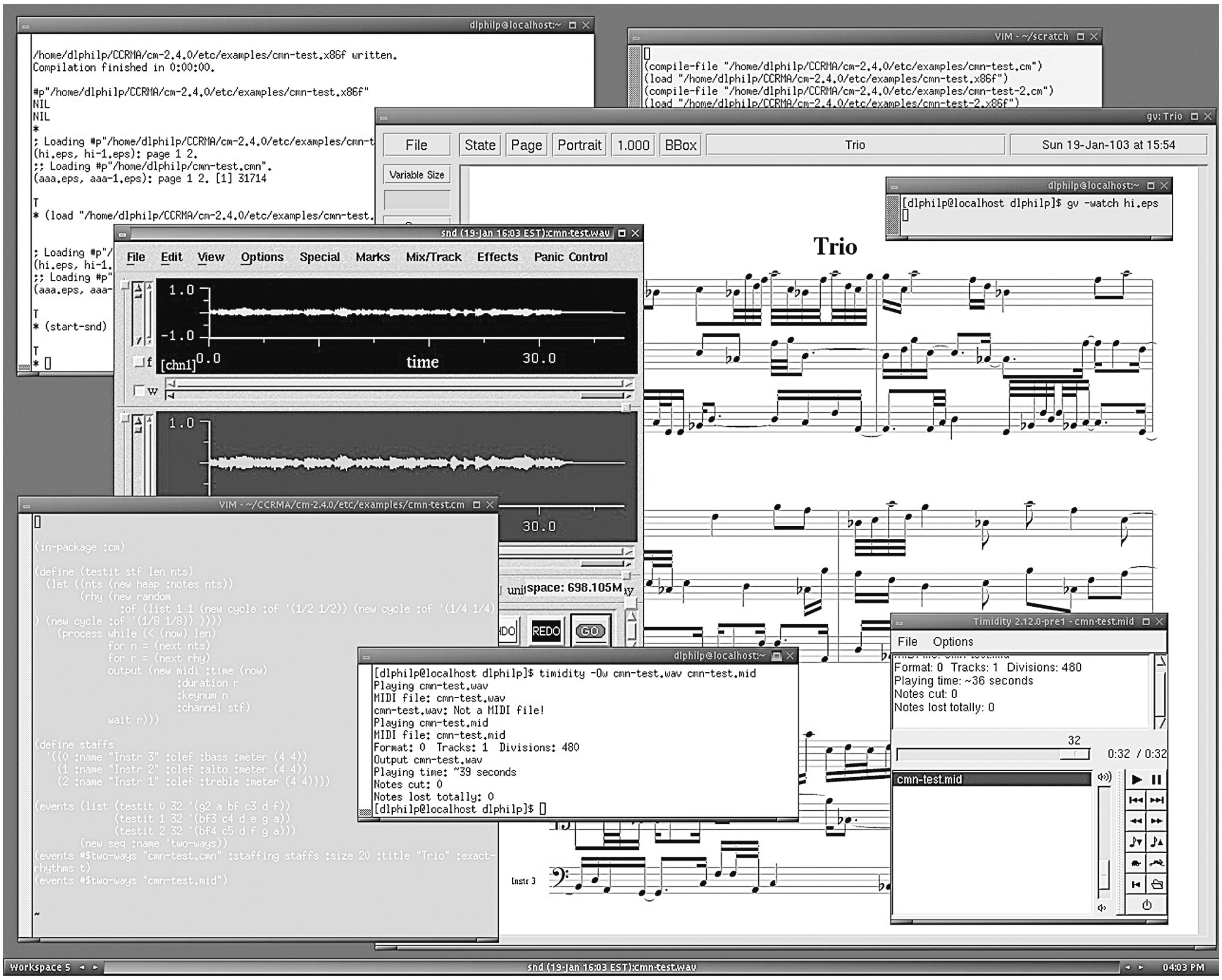


Figure 1. A screen image showing Common Music and Snd.

As expected of a worthy heir to the lineage of UNIX software, Linux shines in its support for sound synthesis packages derived from the seminal Music V. Csound, Cmix, cmusic, Music4C, and Common Lisp Music are all freely available in versions for Linux, all packaged with source code (though not all are licensed under the GPL). The extent of Linux-specific support varies: Csound for Linux has received much attention from its worldwide development community, resulting in support for full-duplex, real-time audio processing, MIDI I/O, X11 graphics, and many of the extensions from Gabriel Maldonado's CsoundAV program (an enhanced version of Csound for Windows). Cmix is available in a specialized version (RTCmix), written by Dave Topper, optimized for exceptional real-time performance. The entire ''Common'' family of music software from Stanford University (Common Lisp Music, Common Music, Common Music Notation, and the Snd sound-file editor) is available with full support for Linux, providing a complete and powerful environment for music composition and sound design. (See Figure 1 for a screenshot.)

Musicians and researchers looking for Max-inspired software will be pleased to find Miller Puckette's Pd (see Figure 2) and IRCAM's jMax (see

Figure 2. An audio/visual Pd patch.

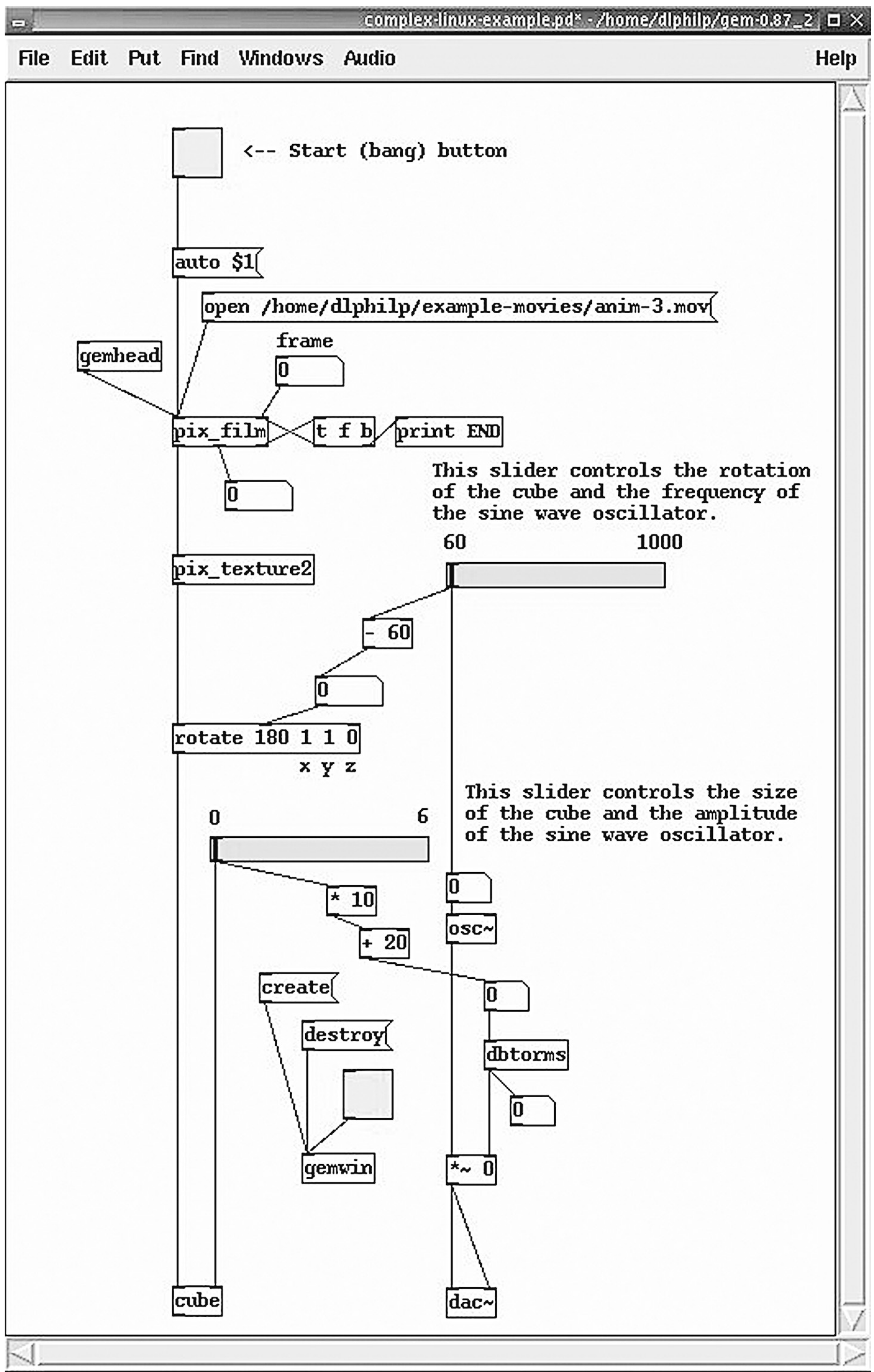


Figure 3). Both systems enjoy very active development and user communities, and both seem to be turning to design philosophies of ever-increasing integrative possibilities. Objects have been designed for real-time video capture and processing, for interfacing with Csound, for animating and controlling OpenGL graphics, and for using software synthesizers and external plug-in architectures such as Steinberg's Virtual Studio Technology (VST) and LADSPA.

Linux sound-file editors deserve special mention. Users migrating from the Windows or Macintosh environments may be disappointed by the apparent lack of an editor in the same class as Cool Edit Pro or Peak, but Bill Schottstaedt's Snd is arguably one of the most powerful editors available for any platform. Its power is largely reserved for users willing to learn the Guile/Scheme programming language, but there seems little it cannot do that is done in Cool Edit Pro (at least at the editing level). Newer sound editors include Conrad Parker's Sweep, the excellent ReZound, Dominic Mazzoni's outstanding Audacity, and the Cinelerra audiovisual editor. Older but still powerful stalwarts include Doug Scott's venerable MiXViews and Richard Kent's Digital Audio Processor (DAP).

At this point a divide exists. The software mentioned so far represents a development tradition based in universities and research centers, but many popular audio software categories (e.g., MIDI sequencers, MIDI patch editor/librarians, and hard-disk recording systems) have been defined primarily by commercial development houses. However, manufacturers of large-scale integrated audio-and-MIDI sequencers and hard-disk recorders (HDRs) such as Cubase or Logic Audio do not provide source code to the end user, nor do highly evolved MIDI sequencers such as Cakewalk. Open-source programmers who wish to write such applications must learn to do so without benefit of an inherited code base, yet that is exactly what has occurred. Paul Davis's open-source application Ardour (see Figure 4) is aimed squarely at users familiar with the ProTools model software for digital audio workstations. Such a task might daunt most programmers, particularly when almost every aspect of the project must be coded from scratch. Nevertheless, Ardour has been designed to reach professional levels of capability and performance, and by mid 2003, it is well on its way to achieving its goals.

The nature of Ardour's development is itself a model of successful Linux audio software development. Although Ardour is essentially the vision and creation of an individual, Paul Davis has acquired a team of talented volunteers to help push the project forward. Contemporary development tools (particularly those provided by the online SourceForge project management environment) are fully employed, and while a high standard of skill is necessary at the core coding level, the project remains open to all who wish to be involved. In a manner similar to Linus Torvald's style, Mr. Davis acts as ''benign dictator'' and final arbiter, ensuring the integrity of his original vision while welcoming new ideas and suggestions from his team members and ordinary users.

Notable current MIDI software for Linux includes the Anthem, MusE, and Rosegarden sequencers. MusE is especially noteworthy for its support of advanced features such as audio sequencing, LADSPA plug-ins, software synthesizers, and other ALSA clients. Rosegarden claims a similar set of features, with a special interest to Csound users: it can save its output as a Csound score (among many other formats). Both MusE and Rosegarden perform MIDI and audio sequencing as well as basic standard music notation, and both are already JACK-aware. Other worthy Linux MIDI software includes the ALSA MIDI patch bay and the excellent JSynthLib universal synthesizer editor/librarian.

Perhaps the easiest way to begin learning about Linux audio projects is to install one of the new turnkey systems optimized for sound and music production: Demudi, Planet CCRMA (from Stanford University's Center for Computer Research in Music and Acoustics, available online at www.ccrma.stanford.edu/planetccrma/software), Turn-Key Linux Audio (lulu.esm.rochester.edu/kevine/turnkey), or AGNULA (A GNU/Linux Audio distribution available from www.agnula.org).

Figure 3. A simple jMax patch.

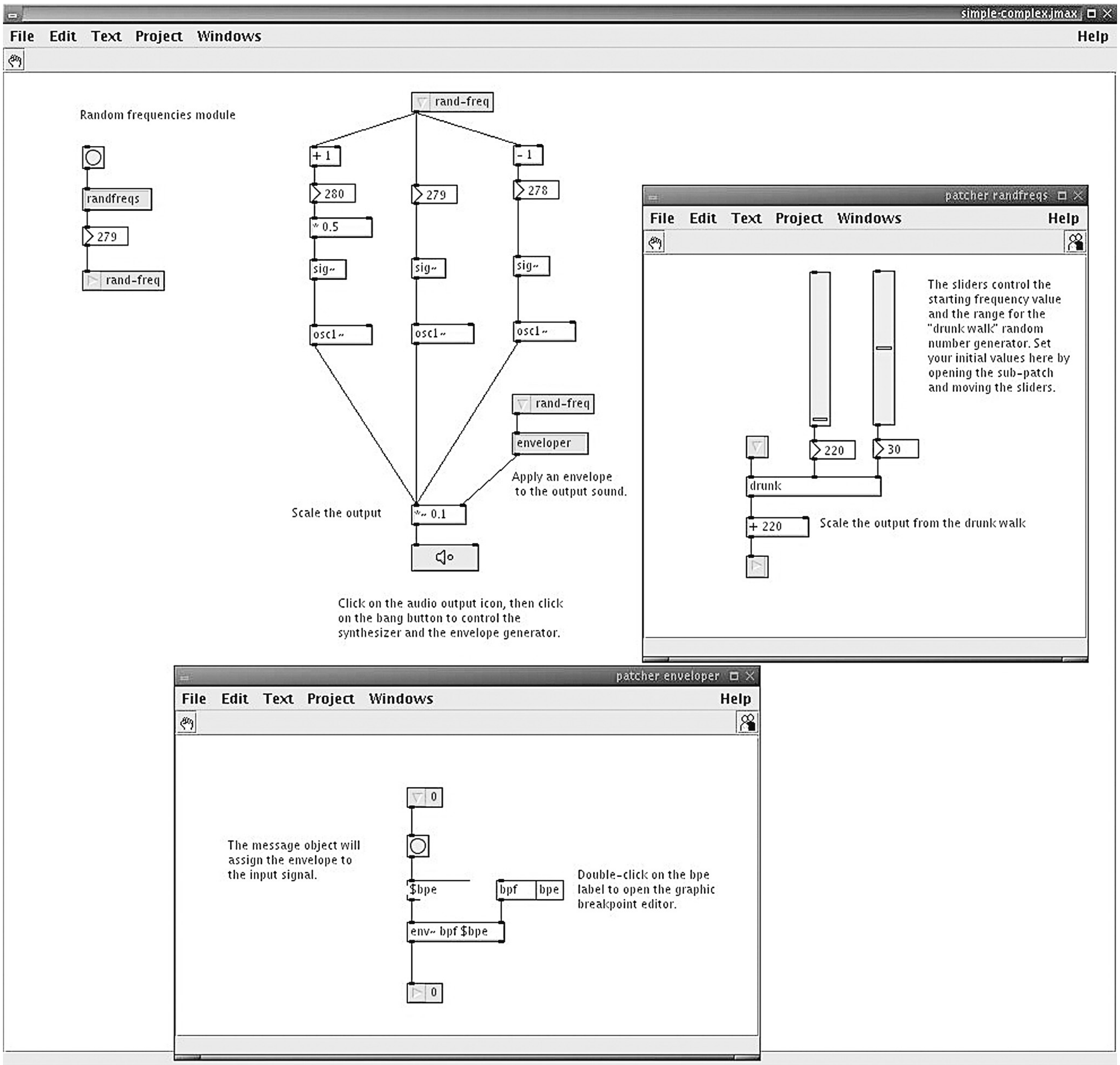
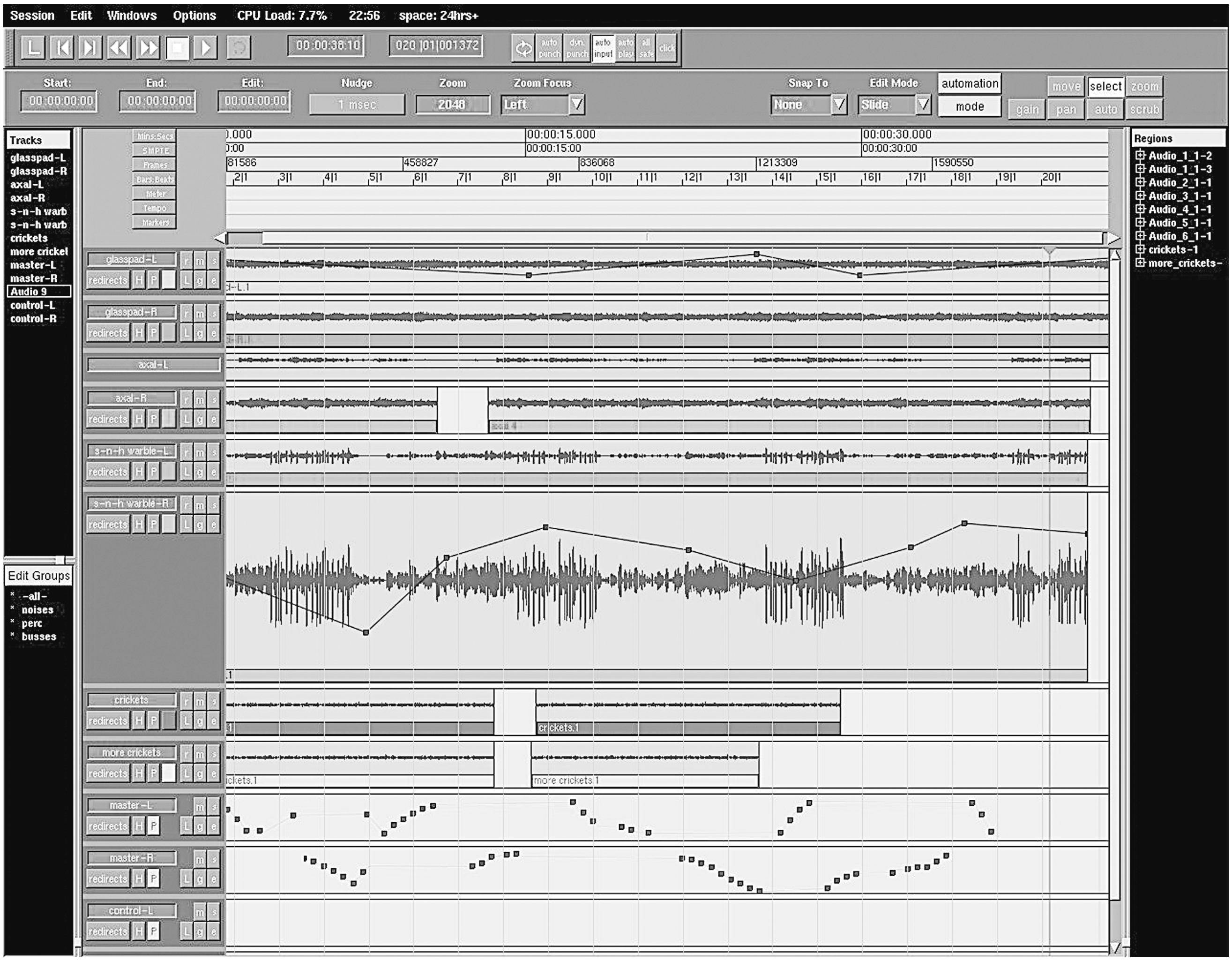


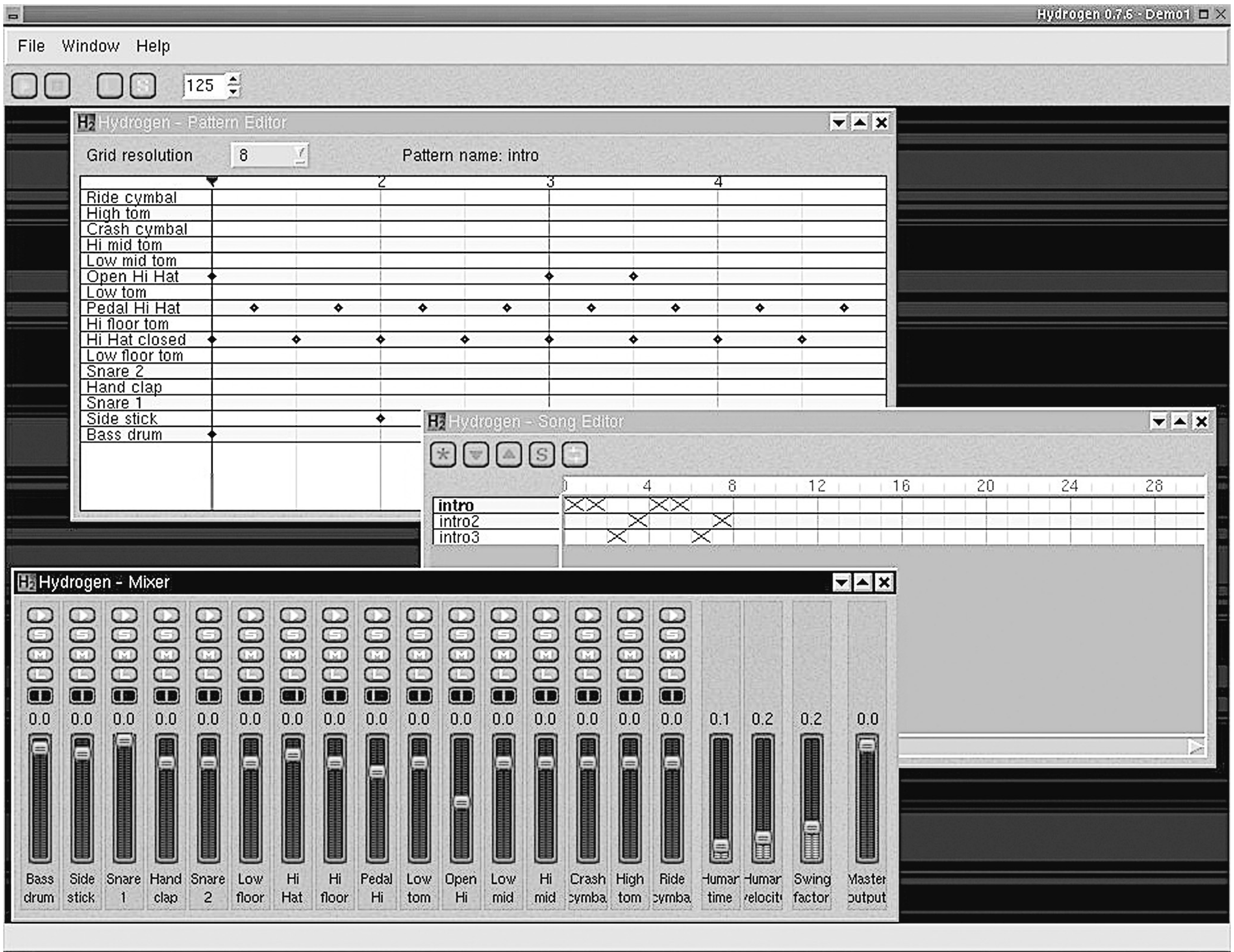
Figure 4. The Ardour editor window.



They provide a low-latency kernel, ALSA drivers, JACK and LADSPA, and a host of sound and music applications. It is important to recognize that these either are full-fledged distributions or require only a simple installation to upgrade an existing system. (For example, Planet CCRMA installs on top of an existing Red Hat installation, and Turn-Key Linux installs on top of a Mandrake installation.) Great efforts have been made to make entry into the world of Linux audio as painless as possible without crippling the power of the underlying system, and these packages are certainly an excellent introduction to Linux in general and its audio software base.

As indicated, the scope of Linux audio software is too broad to delineate in detail here. Other notable categories include applications for working with CD technologies (rippers, burners, players), telephony systems (see especially the Bayonne project), digital disk-jockey (DJ) tools, virtual drum machines (e.g., the Hydrogen drum pattern editor; see Figure 5), MP3 and OGG audio compression software, MOD trackers, software synthesizers (e.g., ZynAddSubFX; see Figure 6), and various net-

Figure 5. The Hydrogen drum pattern editor.

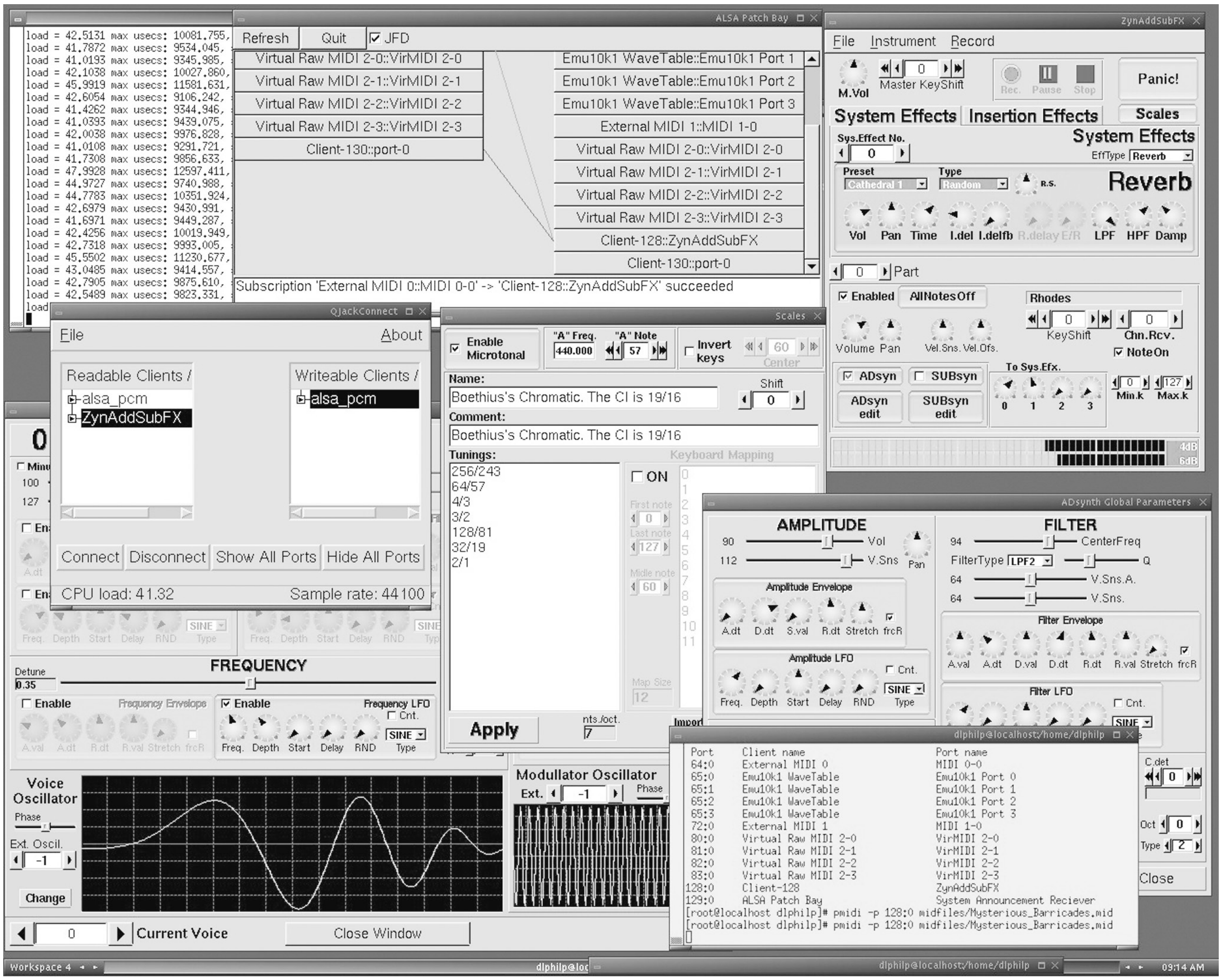


work audio solutions (network sound servers and streaming audio delivery systems). Interested readers are again advised to consult one of the Linux Sound and Music Applications sites for a complete survey.

Commercial Audio and Music Applications

At this time, only one company (4Front Technologies; see www.opensound.com) has survived as a commercially viable entity selling Linux audio software. However, 4Front provides drivers for a wide variety of UNIX platforms, not only Linux. For some platforms, 4Front is the only source for audio drivers, so it must be noted that they do not depend solely upon Linux users for the company's revenue flow. Other notable attempts at commercializing Linux audio software include the Ceres sound file editor (not to be confused with the spectral domain processor of the same name from the Norsk Nettverk for Akustikk, Teknologi, og Musik, or NoTAM), the Ultramaster RS101 rhythm synthesizer, and the Jazz MIDI sequencer. Of those

Figure 6. The ZynAddSubFX software synthesizer.



attempts, only Jazz has survived (as Jazz++, an open-source audio-and-MIDI sequencer project available under the GPL). Ultramaster and Ceres have seemingly disappeared, but commercial Linux audio software may yet occupy a viable market position. Companies such as Steinberg and Twelve-Tone Systems have started to show interest in the Linux audio community, and as more users shift to Linux from Windows or the Macintosh, they will likely demand more of their familiar software for the new platform.

Conclusion: Prospects for Linux Audio

In March 2003, I attended the first conference dedicated solely to Linux audio development, hosted by the Zentrum fur Kunst und Medientechnologie (ZKM) in Karlsruhe, Germany. For three days, Linux audio developers from Germany, France, Italy, Belgium, England, and the USA met to discuss a wide variety of issues related to audio software development at all levels. The conference was a great success, clarifying and resolving some current issues, and defining many other future plans and directions. Outstanding presentations included demonstrations of Ardour, Pd, the Bedeviled Audio System (BEAST), gAlan, Common Lisp Music, and jMax, as well as technical discussions regarding the ALSA system, LADSPA plug-in programming, and the evolution of the AGNULA and Planet CCRMA environments. The high levels of skill and cooperation served to underscore my belief that although there is indeed still a long path ahead, it is getting shorter every day, thanks to the efforts and talents of Linux audio software developers.

Linux audio and MIDI software development moves steadily forward. The Ardour HDR/DAW is already quite sophisticated and is likely to become the ''killer app'' for Linux audio, Kjetil Matheussen's VSTserver has made it possible to use VST plug-ins via LADSPA or Pd, Istvan Varga's CsoundFLTK is a state-of-the-art version of Csound for Linux, and the JSynthLib project supports an ever-increasing number of MIDI synthesizers. Like Linux itself, this software is under constant development by its programmers and continual review by its users. The Linux audio community is steadily expanding, as is the amount and quality of documentation and other tutorial material. Linux itself has become easier to install, with greatly improved hardware recognition and support, and its community of users provides a vast reference resource for ''newbies'' and gurus alike. These factors, in conjunction with the increasing sophistication of its audio components and the availability of turnkey systems, make it clear that Linux is evolving into a platform worthy of serious consideration by sound researchers, musicians, and multimedia artists.