

Getting Linux to produce Music fast and powerful

Hartmut NOACK
www.linuxuse.de/snd
Max-Steinke-Strasse 23
13086 Berlin,
BRD,
zettberlin@linuxuse.de

Abstract

At the LAC 2006 I introduced a plan to build a PC to serve as a Linux Audio Workstation (L.A.W.). Now we have a prototype of this machine, that we would like to demonstrate. The box shall be displayed, tried and reviewed by visitors of the conference to find out, how its concept can yield its intended results. The machine has a set of essential GUI-oriented Linux audio software that works with jackd and is integrated with scripts, presets and templates that allow to start complex scenarios with a single click and comes with extensive user-oriented documentation.

Keywords

usability, integration, suite, interface, workstation

1 Introduction

Musicians often state that they don't know very much about computers, that they only want to use these boxes and that they have neither the time nor motivation to learn how a computer or software works. Proprietary software vendors try to adapt to such attitudes by designing all-in-one applications, such as Steinberg's Cubase, with simple-looking interfaces and automatic setup assistants that leave power usage and fine tuning to the experienced users and hide possible options and - of course - the source code from the "end user" to keep their products under control and consistent.

This is not the way that Linux audio can go, since it is free/libre open source(floss). Free software authors, as well as the distributors, need to develop other ways to achieve usability and a trustworthiness that meets the needs of productive use. This can be done if the whole audio workstation is developed and built as an integrated combination of software **and** hardware components, that fit together and are Linux compatible down to the last screw. The complexity of the software environment, that comes with the way, this software is developed, cannot and should not be simplified to achieve

better usability. But it can be made much more accessible by explaining how it ticks in user oriented documentation and tutorials.

Some may ask:

is this still possible without charging the users for software?

I say: It is! - if we find a way to charge the users for integration and support... it is my belief that the time is nigh to make people invest in free audio software development. People like Fons Adriaensen, Rob C. Buse or Paul Nasca should be rewarded for their work. A reasonable part of the revenue that could be generated with floss-based audio workstations should go directly to the developers of the most important software for these boxes - especially if these developers do not yet receive any reasonable financial support.

The prototype of the L.A.W. is able to prove, that musicians can get not only the same, but very new and exciting results when using Linux for production. So we also intend, to find musicians that are willing to invest in free software development by purchasing a preconfigured hardware system such as the L.A.W. We commit 30 Percent of the profit that is made with each box sold to be donated as a contribution to linux audio developers. Though this can only be a symbolic act for we lack the resources to gain a big market-share, we consider it a start to better funding of Linux audio development.

2 Gathering functionality the UNIX-Way

Most users coming from proprietary tools, such as Steinberg Nuendo or Magix Samplitude, have a rather ambivalent impression of the system when looking at Linux solutions. On the one hand they like tools such as Jamin or SND, but on the other hand they are disappointed by the lack of features that they have commonly come to expect in proprietary applications. The concept that a complex system can be built out of

many little tools that provide each other with functions is not very common amongst users of proprietary systems.

Testing demos of commercial suites on MS-Windows and comparing this way of working with the work in a Linux audio environment I came to the conclusion, that the sometimes uncomfortable diversity and modularity of Linux audio is indeed its strength - as an artist I want to do new and unusual things, I want to experiment, I want to go beyond limits, I want to be free. Jackd provides such freedom by allowing to combine virtually anything one can do with audio on computers. As the tools derive from the free software culture, they are built to cooperate - there are no reasons to avoid integration with tools from others and there are no reasons to provide many features that are provided by others already in a single software suite.

Some may argue that applications like Rosegarden and Muse try to be just that: one-stop applications like Cubase. But to honour reality: has anyone seen significant progress for the wave tracks of these applications in the past 2-3 years? And still those applications have seen significant development indeed - in their special domains as sequencers. Truth is: those applications do not need powerful wave editing features because their users can edit wavefiles to be merely played in them with a plethora of other programs and/or sync with Ardour. The recent Muse and Rosegarden allow calling an external audio editor to edit wave segments and can reintegrate the results - that's the UNIX-way and it is implemented perfectly.

So the L.A.W. does not make the futile attempt to mimic proprietary audio suites but embraces the diversity of free software and tries to give the user a hand to ease the use of the Linux audio tools. This primarily means 2 things:

1. Integrating all the little, cute and powerful free tools to make them work properly together, without limiting their flexibility
2. Providing readable, usage oriented documentation.

3 Not much else matters - reliability or death

If one asks a musician/producer, what could be the most important feature of an audio workstation, there may be different results. But if you ask, if occasional crashes would be an acceptable price for many extra-features they will

consider you crazy. If you work for (paid) hours with a band there will always be a good explanation why a certain feature is missing - if the box freezes and the work is destroyed, not even the most creative producer will find a reason to have this accepted. So to make a producer feel comfortable with a workstation it must be rock-solid. There are 3 prerequisites for this type of comfort:

1. Every part of the suite works with the given drivers.
2. The parts - soft and hard, do not conflict with each other
3. The documentation points the user to bugs and pitfalls that still exist in the software and shows ways to work around them.

Point one is a matter of testing, experience and understanding of predictable user behavior (as far as there is something like predictable user behavior...). Point one and two can be achieved with free components under Linux, but it takes a lot of effort to set it up and there is still no integrated user interface that allows truly intuitive work with the full power of available possibilities without unwanted surprises. The ingredients for a workstation that is both powerful and stable are available for Linux and PC hardware properly supported by Linux is available too. So the first steps to build a Linux Audio Workstation would be:

1. To design a hardware setup for a reasonable price, that fits perfectly together and is completely Linux proof.
2. To develop a set of scripts and consistent templates and presets that allow the user to switch on the full force of a jack-environment with a single click without breaking something.
3. To write complete and up to date documentation with the user in mind.

If Linux software is simply installed as a binary package and automatically set up, then Linux will not be as stable and consistent as is required. Many developers work and build on Fedora or Debian and most of them massively alter/tune their systems - so that users who try to run preconfigured applications on out-of-the-box systems from Novell, Canonical or Mandriva will be confronted with the unpredictable

effects of diversity. No installation program can predict everything and certainly no installer is capable to set an audio card to a single IRQ. So a reliable, full force system for audio can only be set up with direct human intervention. Point three of the prerequisites - the documentation of pitfalls - is a matter of testing, experience and understanding of predictable user behavior (as far as there is something like predictable user behavior...). The goal is, to provide the flexibility of the multitude of combinations, that can be used with Linux audio software and to make the unevitable learning as easy and non-frustrating as possible.

Since we cannot rely on consistency - we need to *use* the diversity and freedom of free software to prove that free development can lead to the same and even better functionality as known from commercial vendors. Whereas the basic system can safely be installed automatically, the important applications (Jackit, Ardour, Muse and some 5-6 more) are compiled on the box with sane `./configure-flags` and the whole system setup is examined, adapted and thoroughly tested by humans.

We did so on the prototype with very good results. Still not everything is as perfect as it should be on a workstation ready for unlimited commercial use. Especially kernel hacking is still an issue, because we want a box that is not only a good audio machine but also usable as a desktop computer. As we use Ubuntu on the machine, we also suffer from Canonical's decision to drop realtime support and other audio related settings in Ubuntu's default kernels due to problems reported with laptops. Using a selfmade kernel or a third party package works well but breaks compatibility with certain drivers and programs like VM-Ware. There is an announcement from Canonical to provide an unsupported `rt-kernel` package with the needed driver support for the upcoming release called feisty fawn. If this will not solve the situation, we need to find another free distribution like 64Studio or Jacklab or help pushing the development of the Ubuntustudio-project, that aims for a multimedia metadistro for Ubuntu. We have a tutorial in the wiki of <http://www.audio4linux.de>, that describes every step to set up the stuff on Ubuntu dapper drake. It provides information on adapting it for the recent Ubuntu edgy eft and links to download the needed files. The L.A.W.-documentation is online and a chapter

that describes the hardware setup is on its way.

3.1 What is being done and how we are different

The user has the opportunity of choosing between 3-4 preconfigured setups:

- Audio workstation (preset configuration with factory support, guaranteed functionality, limited configurability)
- Audio workstation experimental (same preset configuration as above, but with full configurability, limited support and no guarantee - this would also be available for download)
- PC Workstation (Standard home computer with all the power of a Linux desktop system, Internet, office, graphics/layout etc.)
- Rescue system (with direct access to scripts, that reset configurations and replay backups)

The desktop and standard audio system uses XFCE4 as desktop, the experimental audiosystem and the rescue will use Fluxbox. All have the same menu, except the rescuesystem of course (there will be a printed documentation on how to use the rescuesystem from commandline in case X is broken, a Live-CD to rescue the system is planned for later). However, all the scripts, templates and presets can be used in any Desktop environment - templates and presets rely on fitting versions of the respective audio applications and the scripts only require bash. KDE-base must be installed also, since the wrapper scripts utilise `kdiallog` to display messages and `konqueror` is used to show HTML help pages.

There are both integrated hardware solutions with Linux and CD and/or metadistros available for installation out there. The hardware systems are designed and priced for semi-professionals and we don't know of any Linux-based audio solution that is also a decent desktop PC. Our aim is to provide a box that can be built for about 700,- EUR and that can also serve as a desktop/internet computer.

The installable Distros such as Demudi, Jacklab or CCRMA all have one thing in common: they work with binaries, they do little to integrate the apps and they leave the choice and setup of the hardware to the user. All these people still do great work and it is definitely not our

intention to replace any of them, but rather to collaborate with them.

We are not trying to build our own little world, but wish to incorporate things that are already there and glue them together in a useful way. As mentioned before, we primarily deliver simple wrapper scripts, presets, templates and samples. These will work on any distro that has the software installed, which is called by these scripts and can handle these presets etc. On the box that we ship, the same things will run like a charm (no kidding - our concept allows intense testing of the very machine that will be delivered - so unwanted surprises will be seldom...) - if one wants to use the stuff in a similar but different environment, adaptations may be needed.

We follow a paradigm that favours a grass-roots style growth of the project. So we have built a standard desktop PC with an Intel PIV 2.8 CPU 1024MB DDR-RAM and a Terratec EWX 24/96 audio card - rather average real world hardware available for less than 700,- . If we can muster the financial resources until then, we will have a second prototype based on AMD 64 with an M-Audio Audiophile card to bring to the LAC.

We have tested the setup by using it to record several multitrack sessions, composing and manipulating music with midi-driven softsynths and by editing and optimising a noisy-tape-to CD - job for the complete "Ring" by Richard Wagner (plus several similar smaller jobs for string quartet and for rehearsal tapes). We also tested the machine with a Midi-Keyboard/controller with very good results. The setup works well and stable and provides everything we need.

The issues regarding compatibility/stability are solved so far (though it would not be wise, to actually *guarantee* full stability for all needed programs under all conditions...)

3.1.1 The additional

We have built the previously mentioned framework of helping elements, consisting of 4 major components:

- XFCE configuration scripts, which allow access to all needed features in a logical and comfortable manner
- an extensive manual in HTML
- several scripts and hacks to load complex scenarios and to provide additional help

text

- a set of templates for all major applications that also involve the collaboration between them
- about 300 free licensed presets, sounds and patterns

Starting setups of several collaborating applications *could* be trivial - if all involved elements were aware of each other. Today we still face the problem of certain softsynths that can be attached to jackd via their own command line, and others that need to be called up via tools such as *jack connect*. These are not serious obstacles of course, but there is no reason not to address smaller annoyances as well as great todos. The communication between the several processes is particularly critical and often leads to ruin, especially if LADSPA-FX is involved - this should be remarked amongst developers and distributors.

The website <http://www.linuxuse.de/snd> offers downloads of sample scripts and templates plus some help texts. More help is to be found at <http://www.audio4linux.de> and we are also working on an extensive user manual that can serve as an introduction to Linux audio, this can be found at: <http://gnupc.de/zettberlin/law/Documentation/> . In addition to the help provided by the programmers, there are also 3 levels of extra help for the users:

- kdialog popups explaining things that happen, ask the user for needed interaction and point to more detailed help.
- HTML help files for *every* application that explain the basics and how the whole system works. It will be possible to start scripts directly from these files, which will be shown in KDE's konqueror. (Security people may be not that enthusiastic about the help system...)
- an online forum (a simple wiki linked to the forum of audio4linux.de) with daily appearance of at least one developer/help author.

4 Next steps

I would like to present a wish list to the developers and to the user community as well. Developers should improve their documentation and users should start to read it.... . Back in 2003 we had a set of experimental stuff that could

be used but was limited by some crucial weaknesses, especially the lack of stability. Today we have the tools complete and solid enough to start to discover how to deploy them as perfectly as possible. To do this, there must be a vivid community of more than just about 100 users. We do our best to spread word about linux audio and we find new linux audio users every day.

We have combined Ardour, Muse, Hydrogen, AMS, ZynaddSubFX and about 20 tools such as LADSPA, qalsatools etc. into a setup that is powerful and usable for us as experienced Linux users, and we now work intensively on making the setup powerful and usable for everyone that wants to deal with music and computers. We monitor the development of the Linux Audio Session Handler (LASH) constantly and try to encourage developers searching for new goals to join its development. LASH could do the tricks, we now perform to integrate applications much better than the quite clumsy scripts we use now. So we will switch to LASH as soon as it gets ready.

We now begin to give users the opportunity to test the complete Linux Audio Workstation, to give sound tech people and musicians a chance to find out that Linux audio is ready to run for everybody.

5 Conclusions

We believe that free audio software can be an important, powerful way to make Linux visible to the public and thus to make the very concept of collaborative, open and free production of software a success. We not only believe that Linux audio is somewhat usable now and could have a niche - we believe that within 2-3 years it can develop into a superior solution for creative sound people in the same manner as Linux has become a success in motion picture animation/CGI - production.

This can be done if it becomes easier to use, more logically integrated, more stable and more consistent **without** hiding anything of its great opportunities from the user.

At LAC I would like to present our approach to making Linux audio usable for everyone to developers and users as well and to demonstrate the prototype of the Linux Audio Workstation in a workshop and/or as a presentation.

6 Acknowledgements

Our thanks go to ...Paul Davis, Werner Schweer, Paul Nasca, Chris Cannam and to all the other great Linux audio developers out there, to the people at ZKM and Linuxtag that allowed us to present our project at Karlsruhe and Wiesbaden in 2006, and to the people at www.audio4linux.de (especially linux-chaos/Olaf Giesbrecht), jacklab.de (Metasymbol/Michael Bohle, www.mde.djura.org (Thac and Ze), ccrma.stanford.edu/planetccrma and all the other websites, mailing lists and BB's where Linux audio springs to life...