

Daniel James talks to developers working on the LinuxSampler project



Breaking out of the loop

Digital sampling in music is now more than twenty-five years old, but for many of those years the technology has been associated with obviously artificial results. Samplers began widespread use as specialised embedded hardware from companies like Akai, with samples of a few seconds the maximum possible; anything longer required rapid looping, so these devices played their part in the creation of several genres of highly repetitive dance music.

Gradually, software samplers running on general-purpose PCs and Macs took over. Sampling has now reached the stage where it can provide relatively naturalistic emulations of real musical instruments, used for both computer playback with MIDI, and human performances when the real instrument is not available, or is simply impractical.

Natural-sounding samples present a number of technical difficulties, since sample sets can run into hundreds of megabytes, or even gigabytes, per instrument. These are generally too large to be played from RAM, so they have to be streamed directly from disc or over the network. One of the original disc-streaming samplers was the proprietary GigaSampler for Windows, created by Nemesys and bought up later by Tascam. Now known as GigaStudio, the success of this software sampler meant that many large sample sets were recorded in its native .gig format.

In the world of free software, the LinuxSampler project has been underway for several years, and has recently made its first official public release.

Benno Senoner is the project founder; he lives in Italy, near the border with Austria. "Using a PC to make a virtual instrument that provides both high quality and is playable in real time is quite a challenge, since you need to squeeze the maximum from the hardware. A PC can be much more powerful than embedded solutions because the pace of innovation in the PC area is much faster than in the embedded space, and of course the cost per performance is lower too."

"I started LinuxSampler as a proof of concept to see what could be done with Linux. Thanks to great coders like Christian Schoenebeck, Rui

A patch for just one instrument can easily be larger than 2GB

Nuno Capela, Vladimir Senkov, Andreas Persson and others, LinuxSampler was turned into a powerful sampler which can compete neck and neck with commercial software samplers. I think over time, thanks to the power of the open source community, it will perhaps even be able to surpass them in any area - only time will tell."

SAMPLING NUANCES

"Since streaming the samples directly from the disc allows for almost unlimited sampling space, this means you can sample many of the nuances of natural instruments. For example, you can avoid looping samples and record long sustained notes. You can record dozens of velocity layers per note and sample each note chromatically so the sampled instrument is almost indistinguishable from

the original one. The fall of the sample memory barrier allows for more expressive playing of instruments. For example, you can have a string sample where you switch the playing style in real time, going from staccato to legato. The possibilities are endless."

As Senoner explains, LinuxSampler is not merely a clone of GigaSampler. "LinuxSampler was designed to be modular: multi-engine and multi-format. The Gig engine was implemented first just because there is lots of sample material available in that format; a sampler without samples is of limited use. We plan to make our own open, extensible sampler format but also plan to support legacy formats like Akai, and others as well."

Christian Schoenebeck is based in Germany. He joined the LinuxSampler project in late 2002, initially porting the libakai library for the Akai sample format to Linux, and writing libgig for GigaSampler support. He began development on the current LinuxSampler code base in mid 2003. "The reason that we still only support the GigaSampler format at this point is that we first wanted to finish accurate support for that format - but it seems we finally reached that goal. The next goal is to improve efficiency; once that is done, we will add other formats and features as well."

Schoenebeck argues that despite the technical hurdles they create, large sample sets offer significant advantages over traditional samples of a few megabytes, or sounds created within the computer by additive synthesis. "With small samples you need to define loops and tweak articulation settings very well in order to achieve good results. But at the end, no small

patch of a natural instrument can compare with a relatively large patch. Simulating natural instruments accurately with a synthesizer is a very hard task, because it implies quite complex models and needs a lot of horsepower."

"There are already a lot of very good, large sample sets for all kind of natural instruments. Today a good piano sample set contains full length samples of all keys, typically about 30 seconds each, and each key is sampled with various velocity values to record the full range of the physical resonance behaviour. That way a patch for just one instrument can easily be larger than 2GB. Even for today's technical circumstances that's too much for loading everything into RAM. Consider that you not only want to be able to play one instrument, but perhaps a complete orchestra at the same time."

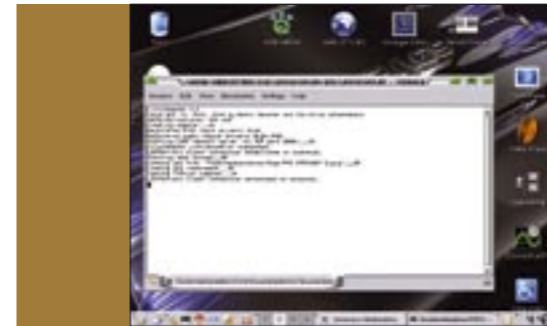
control headless LinuxSampler boxes from a Windows or OS X box with it. Besides that, extensive SMP and network cluster support is planned to allow an automatic load balance of a huge sampler server system, still looking to the outside world like a single sampler."

Rui Nuno Capela lives in Portugal, and has been working on a GUI for LinuxSampler since the autumn of 2003. "It was just three years ago that I jumped on the Linux audio development bandwagon, mainly due to my long-term interest in electronic music. Thanks to the free software paradigm, I could ditch my Windows-based home studio for good, which I only regret not doing before."

The QSampler front-end is written in C++ using the Qt3 toolkit, wrapping the reference interface for LSCP via liblscp. "At the moment, almost every aspect of the LinuxSampler control

processor! I knew there must have been another option, and I was willing to spend some time learning about digital music, as it was a completely new area for me. This was when I found LinuxSampler."

Senkov believes the first public version of LinuxSampler, 0.3.1, is ready for users to download and try out, rather than just being a developer release. "As with many open source packages, users may need to be slightly on the advanced side, especially when it comes to building from the source. Older GCCs may have some optimisation issues that we don't handle well, for instance, but if everything is pre-built, using the sampler should not be difficult. I hope that with this release we'll grow the userbase, and get valuable feedback on what features people would like to see. That will help our project



LinuxSampler is a server application, controlled by the LSCP protocol



QSampler is a Qt front-end for controlling LinuxSampler

THE LINUXSAMPLER INTERFACE

LinuxSampler can be controlled over the network using LSCP, the LinuxSampler Control Protocol. Senoner believes this will create an opportunity for free software in the recording studio, by allowing headless Linux boxes to be dedicated to sample streaming. "It will drive the cost of sampling down significantly, since all you need is a bunch of cheap Linux boxes equipped with Ethernet. There is already a similar product on Windows, but given Linux performance in the networking and real time areas it would certainly perform much better. The goal is to make a LinuxSampler cluster remotely controllable by a standard DAW machine, and provide a VST or Audio Units plugin interface to see the LinuxSampler cluster as a giant virtual instrument."

Schoenebeck explains that LinuxSampler is essentially a server application, decoupled from any user interface. "It offers a network interface based on TCP which allows the user to control it locally or from any arbitrary place. The network protocol is ASCII based, so it's possible to write a frontend for LinuxSampler in any programming language and for any OS. Rui Nuno Capela wrote a frontend for LinuxSampler called QSampler which is based on Qt and can be compiled for all major OS's. So you could also

protocol is already covered. In this respect, QSampler is perfectly usable for controlling LinuxSampler, as far as current server implementation goes. You can set up multiple sampler channel strips, load instruments, or configure audio and MIDI devices. Most importantly, you can manage all this as sessions, which QSampler loads and saves as LSCP script files. One thing is worth noting, however: QSampler is not a sample set file editor, nor was it designed as such - at least as far as LSCP is concerned. So don't expect to create your own sample library with it."

THE FIRST PUBLIC RELEASE

Originally from Russia but now based in the US, Vladimir Senkov is one of the developers of the LinuxSampler engine. The son of professional musicians, he attended a prestigious music school, but decided to become an engineer instead of a pianist. "I spent about eleven or twelve years of my (non-existent) childhood practising the piano for eight hours a day."

"After about ten years of not playing any piano at all, I decided to get myself a MIDI controller. Getting a 'hardware' synthesizer just didn't make sense to me. Being an engineer, I know they are computers on the inside, just specialised - with less memory and a slower

leaders to shape the future of LinuxSampler. While the focus of the first release was trying to get accurate .gig playback, LinuxSampler certainly won't stop there."

"We've had the most amazing and helpful feedback from folks who took LinuxSampler for a test drive, joined the mailing list, shared their ideas and sometimes even implemented them!" Schoenebeck points out the project could benefit from help in several areas. "We need testers, skilled graphical artists, people who write documentation and somebody to maintain the web page. Especially, we need more active developers for the LinuxSampler core and frontends. Senoner agrees: "People can help create new engines, GUIs of course, new sampling formats, new audio/MIDI interfaces, networking, optimisation or building royalty free sample libraries which the LinuxSampler community can use."

Key Links

- LinuxSampler homepage
www.linuxsampler.org
- SourceForge page
sourceforge.net/projects/linuxsampler
- QSampler homepage
qsampler.sourceforge.net
- Contributed .gig format samples
www.worrasplace.com