Usando plugins de áudio VST no Linux

Enfim, efeitos!

credits of the images

O áudio digital profissional foi outrora campo restrito a sistemas proprietários como o Apple Macintosh, mas o Linux já avançou muito nessa área. Agora você pode até mesmo penetrar na vasta reserva de plugins de áudio VST para sistemas Windows diretamente de seu estúdio movido a Linux.

por Dave Phillips

No mundo do software para áudio digital, um componente adicional que estenda as capacidades principais do programa é conhecido como *plugin*. Um *plugin* pode oferecer um efeito adicional como uma câmara de eco, pode ser um instrumento MIDI virtual como um sintetizador General MIDI multitimbral ou uma interface para um seqüenciador MIDI externo.

Um aplicativo hospedeiro e um *plugin* em potencial devem concordar em como, o que e quando eles se comunicarão. Se o programa e o *plugin* concordarem a respeito de uma interface padrão de programação (*Application Programming Interface* – API), eles cumprem muito bem seu papel. Uma API pode ser específica do aplicativo, como as APIs dos *plugins* para o Gimp e o XMMS, ou pode ser projetada de forma padronizada, permitindo que um mesmo *plugin* funcione, sem modificações ou recompilações, em uma grande variedade de programas.

O projeto de um aplicativo que acomode uma interface de *plugin* padronizada é algo que tem grande apelo junto aos programadores. Os desenvolvedores dos programas principais podem se concentrar em criar núcleos mais robustos e flexíveis, enquanto os desenvolvedores de *plugins* podem cuidar apenas das funções específicas de seus *plugins* na área de seu interesse, sem se preocupar com os problemas de adaptação para uma porção de aplicativos diferentes. Claro que os usuários sempre se beneficiam desse tipo de sinergia.

Nos mundos musicais do Windows® e do Mac, a API de *plugins* VST, desenvolvida pela empresa alemã Steinberg, tornou-se a interface padrão mais aceita. A API VST (*Virtual Studio Technology*) chegou ao mercado no final dos anos 90; o atual padrão VST2 surgiu em 1999. Sua imensa popularidade entre desenvolvedores e usuários pode ser avaliada com uma simples busca no Google por *VST plugin*. No momento em que eu escrevia este artigo, o buscador mostrava "cerca de 187.000" ocorrências. É interessante notar que uma busca por *free VST plugin* (*plugin* VST grátis) resultou em quase 95.000 ocorrências.

O mundo Linux tem suas próprias APIs de *plugins* de áudio (veja o **quadro 1: LADSPA**). Porém, apesar do sucesso – e da excelência técnica – do LADSPA e outras alternativas no Linux, o volume razoável de *plugins* de áudio disponíveis através da API VST significa que os usuários de Linux interessados em áudio encontram por vezes situações em que podem realmente precisar rodar um *plugin* VST no Linux.

Este artigo mostra como configurar seu Linux para poder usar o vasto acervo de *plugins* VST, originalmente desenvolvidos para uso em sistemas Windows®, graças ao *vstserver*, desenvolvido a partir de tecnologia criada para um outro projeto Open Source, o *Wine*.

VST/VSTi

Os *plugins* VST tornaram-se componentes muito bem-vindos em estúdios profissionais de gravação e produção de som, sejam baseados em Windows® ou calcados nos Macs. Alguns *plugins* de altíssima qualidade (e preço) são, hoje, pedras fundamentais na produção sonora profissional. Além deles, muitos *plugins* ainda de alta qualidade mas de baixo preço ou mesmo gratuitos permitem que estúdios menores e músicos em seu estúdio caseiro possam produzir peças de um nível impressionante – às vezes até rivalizando com o dos caríssimos grandes estúdios.

A API dos *plugins* VSTi estendem a implementação VST por permitir que o *plugin* seja um instrumento (ou seja, um sintetizador, *sampler* ou seqüenciador) e não apenas um modificador de som. Da mesma forma como na API do VST, os *plugins* VSTi tornaram-se componentes padrão em programas de música e tratamento sonoro nas plataformas Windows e Macintosh. Como mostrado na **figura 1**, um instrumento VSTi típico é um sintetizador por software, "tocável" via protocolo MIDI, com um controle paramétrico em tempo real (manual ou via MIDI) e, possivelmente, com saída multicanais.

Qualquer software moderno de produção de áudio para Windows® ou Macintosh está preparado para receber *plugins* VST e VSTi. O número de aplicativos que aderem ao padrão é simplesmente grande demais para que possamos enumerá-los aqui – o Google é mais eficiente nisso. Entretanto, os últimos itens a serem incluídos nessa lista são de nosso especial interesse: finalmente os *plugins* VST/VSTi podem ser usados nos aplicativos de som e música no Linux.

A possibilidade de usar *plugins* VST e VSTi no pingüim foi desenvolvida primeiramente por Kjetil Matheussen no *NoTAM*, um centro de pesquisas sobre acústica e música da Noruega. O esforço inicial de Kjetil resultou no promissor *vstserver*, o marco inicial de uma arquitetura cliente/servidor para a execução de *plugins* VST. O servidor é baseado na capacidade da onipresente biblioteca *Wine*, uma parte do ainda inacabado projeto *Wine*, cujo objetivo é recriar a API do Windows® dentro do Linux. Kjetil também criou dois clientes para uso com o *vstserver*. O primeiro foi um "objeto" para o *Pure Data*, o conhecido *Pd*, um ambiente para sintetização e composição musical. O segundo foi um *plugin* para o LADSPA (*Linux Audio Developers Simple Plugin*) [1], uma espécie de "VST livre" para o Linux. O *plugin* VST do LADSPA age como um programa hospedeiro para os *plugins* VST de verdade e pode ser usado em conjunto com os *plugins* LADSPA nativos.

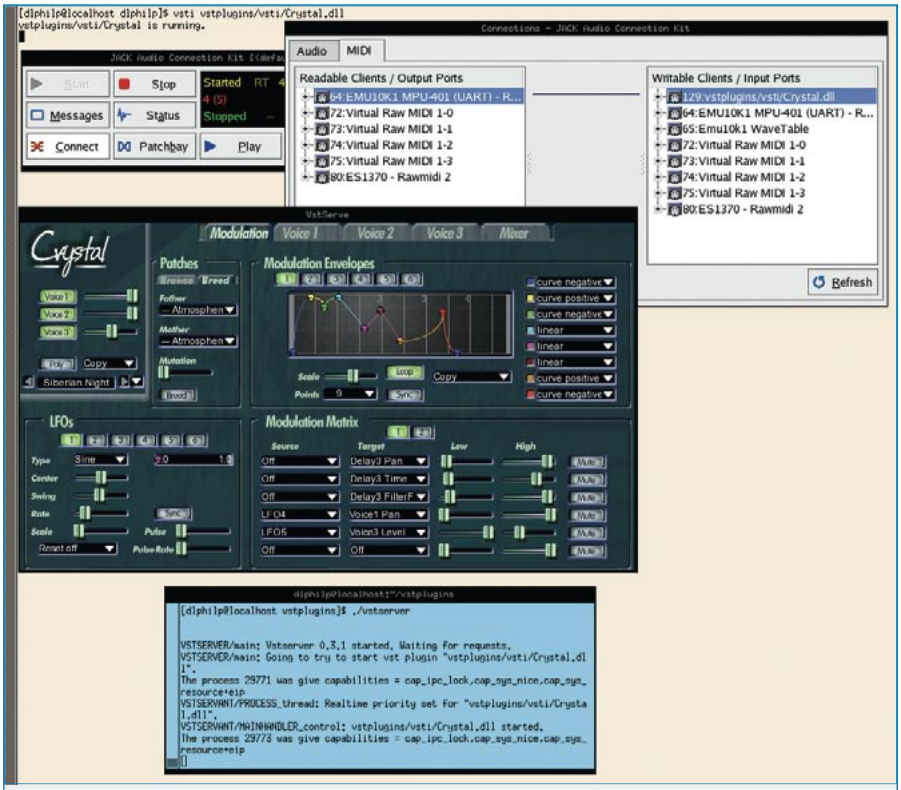

Figura 1: O plugin Crystal VSTi sendo controlado por um teclado externo.

Recentemente, Kjetil desenvolveu um terceiro *plugin*, que permite o uso de instrumentos VST (VSTi). É claro, poderíamos usar *plugins* de instrumentos nativos no padrão DSSI [2], mas ter milhares de instrumentos VSTi à disposição é altamente desejável. Portanto, graças ao *vstserver*, é possível rodar efeitos e instrumentos VST no Linux.

Para instalar o ambiente *vstserver* em seu sistema, siga a bolinha saltitante:

- Baixe, compile e instale o pacote do *Wine* modificado por Kjetil, disponível no site [3];
- Baixe, compile e instale os pacotes *vstserver*, *k_vst~*, *ladspavst* e *vsti* mais atuais que encontrar, eles também estão disponíveis em [3];
- Ajuste a variável de ambiente `VST_PATH` exatamente como descrito na documentação do *vstserver* – só assim, o servidor saberá onde encontrar seus *plugins* VST;
- Inicie o servidor a partir do diretório onde estão os códigos fonte;
- Execute um dos clientes. O *Pd* é necessário para o `k_vst~`. Para o *plugin* VST LADSPA, é preciso um aplicativo hospedeiro LADSPA. Uma boa pedida é o velho editor de áudio *Snd*, modificado por Kjetil como prova de conceito. O cliente *vsti* é um aplicativo completo – não precisa de um hospedeiro – do qual falaremos mais adiante.

Cada pacote inclui instruções bastante detalhadas sobre a compilação e a instalação do software. Por sinal, quem quiser usar o sistema de Kjetil deve baixar e compilar a versão modificada do *Wine*, disponível no site do *vstserver*. O servidor é muito sensível à versão do *Wine* instalada. Outras versões que não a de Kjetil podem não funcionar ou causar instabilidades no servidor.

Um lembrete: conforme indicado na documentação do *vstserver*, a segunda etapa (compilação) requer o pacote de desenvolvimento da Steinberg, o VST SDK (*System Development Kit*). Apenas dois arquivos são necessários para compilar o servidor. Embora o SDK seja gratuito, não é de livre distribuição e deve ser

Quadro 1: LADSPA

No início do ano 2000, o programador Richard W. E. Furse propôs à comunidade um esqueleto do que seria uma API simples para *plugins* de áudio no Linux. Sua proposta foi estimulada e grandemente melhorada por diversas discussões na lista *Linux Audio Developers*. Essas discussões levaram, enfim, à atual LADSPA – *Linux Audio Developers Simple Plugin API* [1] (ou seja, essa história inicial de "interface simples" continuou no nome definitivo).

É importante frisar a palavra <I>Simples<I> do acrônimo. *Plugins* LADSPA não aspiram pelo mesmo nível de interação complexa com seus aplicativos hospedeiros como seus irmãos proprietários, os *plugins* VST. Entretanto, esse "simples" não quer dizer "débil": há *plugins* LADSPA em profusão, todos poderosos e de excelentes qualidade.

Parte da simplicidade do LADSPA está na maneira com que trata da interface com o usuário: simplesmente... não trata! O programa hospedeiro, o sistema operacional e o ambiente gráfico em uso devem cuidar de coisas como desenhar janelas e criar botões. Portanto, ao contrário dos plugins VST, que têm cada um a sua interface gráfica definida, os plugins LADSPA têm uma "cara" diferente para cada programa em que forem usados – embora as funções e controles sejam os mesmos. Por serem assim tão simples e não dependerem de uma interface específica, os *plugins* LADSPA podem ser usados até mesmo com aplicativos em modo texto.

Outro resultado bem-vindo da simplicidade do LADSPA foi sua facilidade de integração com aplicativos já existentes. A API foi rapidamente adotada por desenvolvedores e, nos últimos quatro anos, a tecnologia LADSPA tornou-se praticamente obrigatória em programas de áudio e música para Linux. A variedade de aplicativos que adotaram a API inclui gravadores multipistas em disco rígido, processadores digitais de áudio, seqüenciadores MIDI, sintetizadores em software, editores simples de áudio e mesmo reprodutores de multimídia.

A chamada *responsividade em tempo real* dos plugins LADSPA é normalmente muito boa, mas lembre-se de que depende de um kernel do Linux preparado para baixa latência e um subsistema de som moderno como o ALSA ou o JACK. Os desenvolvedores do LADSPA aceitaram bem a simplicidade da API, que sofreu pouquíssimas mudanças substanciais desde que foi lançada a versão 1.0. Uma extensão digna de nota é a *Resource Description Framework* (RDF), um excelente mecanismo para categorizar os *plugins*, ajustar os valores padrão e permitir *presets* adicionais. A **figura 2** mostra o RDF a todo vapor com uma lista de *plugins* LADSPA na máquina virtual de percussão *Hydrogen*.

Se você estiver usando uma distribuição Linux otimizada para o áudio, a LADSPA já estará completamente instalada e devidamente configurada. Os usuários do Mandrake podem encontrar o SDK da LADSPA e uma respeitável coleção de *plugins* no site do *Thac* (veja a referência [10]). Mas não tema, pequeno gafanhoto: a LADSPA mantém a promessa de simplicidade até mesmo durante sua instalação e configuração.

Obtenha o código fonte de [1], descompacte em seu diretório pessoal e entre no recém-criado diretório `ladspa_sdk`. Leia o arquivo `README`, siga suas instruções e entre no diretório `src`. Edite o `makefile`, rode o comando `make` para compilar o SDK, depois torne-se *root* e rode `make install`. Pronto! Estamos aptos a instalar e usar alguns *plugins* LADSPA.

Verifique alguns dos links em [3]. Há lugares que você vai querer visitar: a coleção de *plugins* de Steve Harris (não, não é o baixista do Iron Maiden...), o "cinto de utilidades" TAP – Tom's Audio Plugins – de Tom Szilagyi, os excelentes filtros de Fons Adriaensen, os brinquedinhos de Tim Goetze e Mike Rawes... Bem, é melhor tentar todos mesmo...

A **figura 3** mostra o *TAP Reverberator* aberto e sendo aplicado a uma trilha no editor de áudio *Audacity*.

O Linux também pode usar outras tecnologias de *plugins* de áudio. Uma delas é a *Multimedia Applications Integration Architecture* (MAIA), de David Olofson. O MAIA foi uma tentativa de resolver as insuficiências do LADSPA. Desenvolvido como uma API genérica e multiplataforma, tinha ênfase especial em sistemas Unix. Mas ai de mim: a API do MAIA não encontrou seu nicho entre os desenvolvedores; talvez por isso mesmo não tenha visto nenhum desenvolvimento desde 2001.

A incursão mais importante no sertão dos *plugins* para Linux vem do desenvolvedores Chris Cannam (autor do *Rosegarden*), o já citado Steve Harris (*plugins* SWH LADSPA) e Sean Bolton. Seu DSSI – *Disposable Soft Synth Interface* (interface descartável de sintetizadores em software) foi projetado para ser um "LADSPA para intrumentos", como informado no site oficial [4]. Trocando em miúdos, o DSSI está para o LADSPA assim como o VSTi está para o VST. Foi criado para resolver um sem-número de problemas existentes na implementação de sintetizadores em software para Linux, especialmente no que toca ao controle via protocolo MIDI. E, além de tudo, uma surpresa agradabilíssima: a interface oferece uma "ponte" para rodar *plugins* VSTi! Até o presente momento, o DSSI foi implementado apenas no seqüenciador *Rosegarden*; se ele será ou não adotado por mais desenvolvedores é uma incógnita. Em minha opinião, é uma API promissora e muito útil. Estou neste momento convocando **todos** os desenvolvedores e usuários ligados a áudio no Linux a acessar o site oficial do DSSI e ler toda a documentação que há ali, especialmente o RFC (*Request For Comments*) sobre o assunto. Vamos lá! Emitam suas opiniões e mostrem algum código!

A **figura 4** mostra o *Rosegarden* rodando com uma instância do *xsynth* de Sean Bolton, uma prova de conceito sobre sintetizadores DSSI.

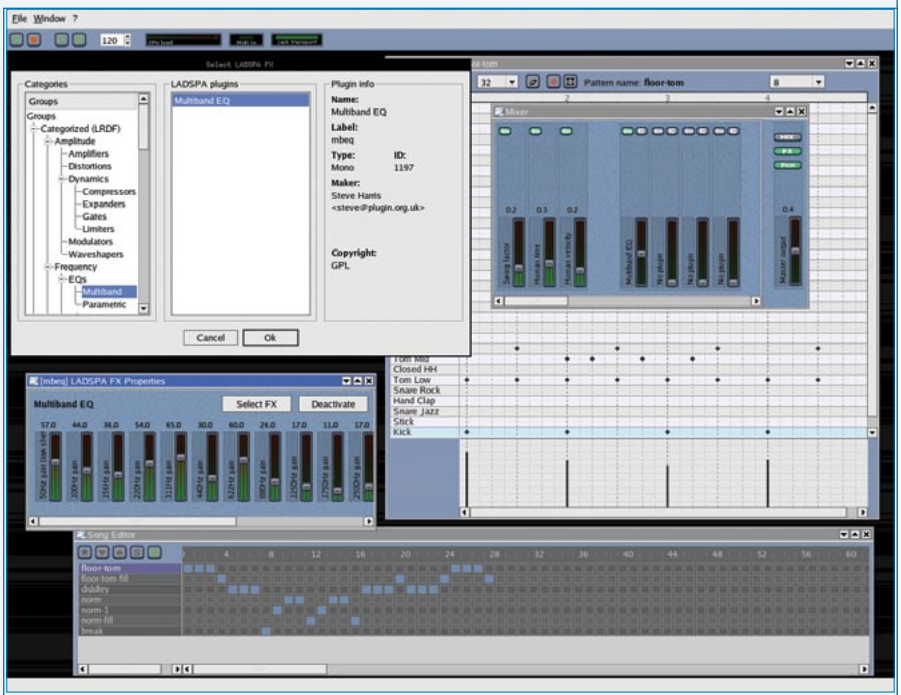

Figura 2: *Plugins* LADSPA listados no *Hydrogen*.

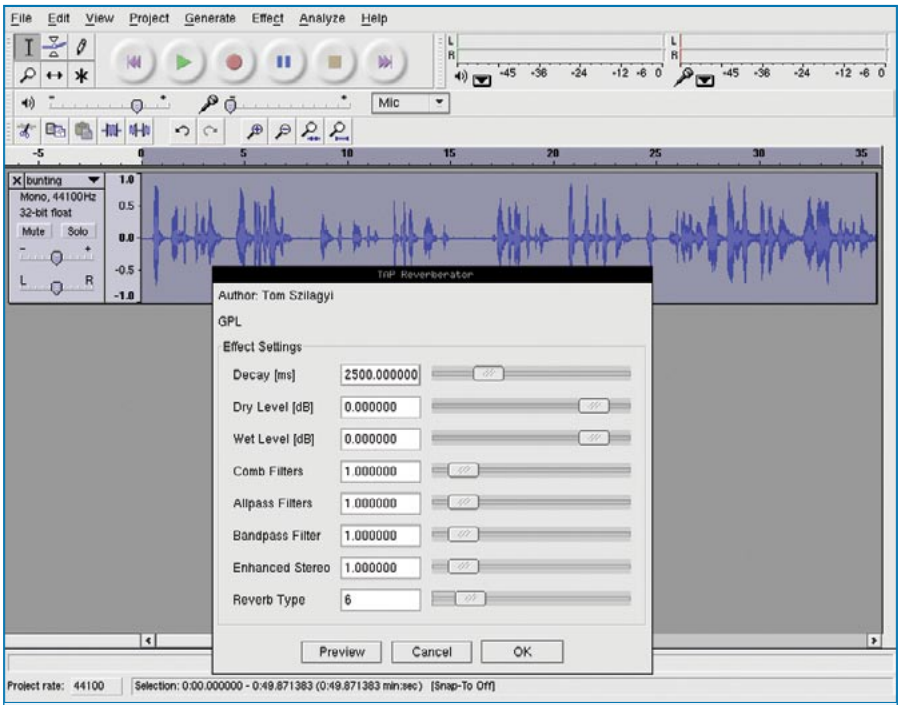

Figura 3: O *TAP reverb* no *Audacity*.

Quadro 2: Pure Data, o famoso *Pd*

O *Pd* (*Pure Data*) é nada mais que a excelente linguagem gráfica de Miller Puckette para a criação de sistemas de síntese complexa de áudio e processamento de vídeo. Os componentes de áudio e vídeo (A/V) são conectados por uma "fiação virtual" para criar um "banco" *Pd* – é possível inclusive enviar esse padrão para os dispositivos de áudio e vídeo. O *Pd* usa ALSA e JACK como subsistemas de saída de áudio. Gráficos em 3D são obtidos por meio da biblioteca GEM.

baixado à parte, diretamente do site da Steinberg (www.steinberg.net). Além disso, se o caminho até os plugins VST estiver configurado corretamente, mas o servidor *vstserver* se recusa a encontrá-los, tente fazer um link simbólico para o servidor e o objeto `vstservant.so` no mesmo diretório determinado pela variável de ambiente `VST_PATH`.

Servidor e cliente

O *vstserver* [5] foi projetado para "escutar" requisições de um aplicativo cliente. Como já mencionei na seção anterior, Kjetil Matheussen criou dois clientes para o *vstserver*:

- um "objeto" para o ambiente de síntese e composição *Pure Data*, também conhecido como *pd*;
- um plugin LADSPA que age como hospedeiro para *plugins* VST.

Ambos os clientes merecem explicações mais aprofundadas, continue lendo para saber mais.

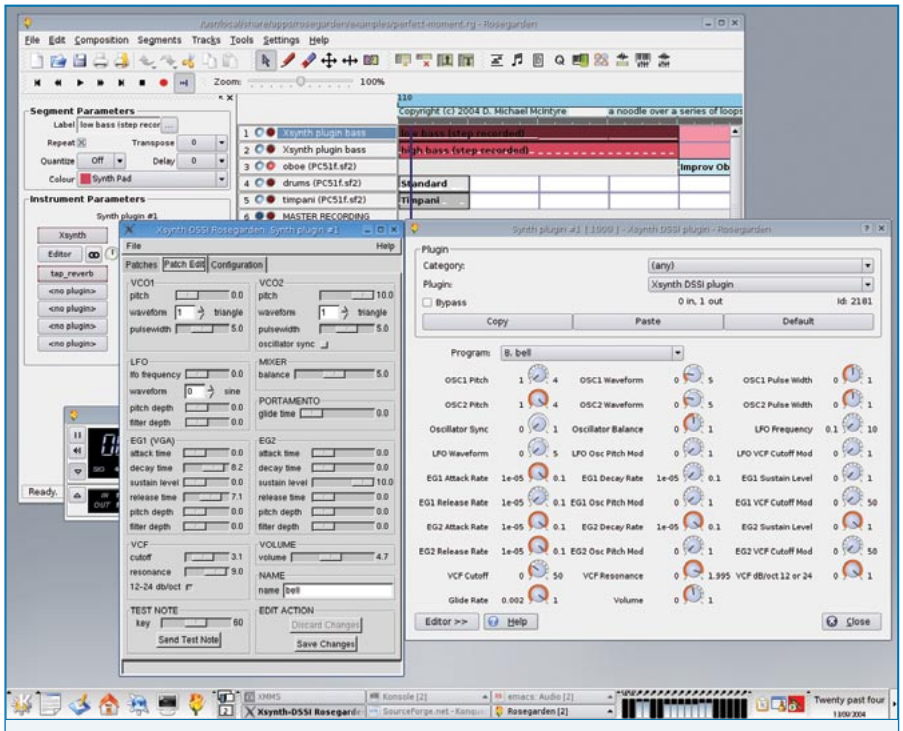

Figura 4: O *DSSI* do *Rosegarden* em ação.

Pd e o objeto *k_vst~*

Inicie o servidor com o comando `vst-server` em um terminal. Você verá uma breve mensagem indicando que o servidor está pronto para receber dados dos clientes. Se você possuir o maravilhoso *Pd* [5] instalado em seu sistema, inicie-o com o comando a seguir, que adiciona o objeto *k_vst~* a suas funções internas:

```
# pd -lib k_vst~
```

Estamos considerando que o comando foi emitido em seu diretório pessoal (dentro de `/home`). Se for rodado em qualquer outro lugar é preciso incluir o caminho completo para o objeto.

A **figura 5** mostra um *plugin* VST de efeito (modificador de som) utilizado dentro de um "banco" *Pd*, que roteia o sinal de entrada pelo objeto `k_vst~ object` (ou seja, o *plugin* VST) e, depois, joga o som modificado em sua placa de som. O *plugin* VST pode ser operado da mesma forma que no Windows® ou no Macintosh; os parâmetros dos efeitos são ajustados simplesmente movendo os controles deslizantes, girando os botões e ligando ou desligando chaves.

Detalhes sobre o "banco" citado podem ser encontrados em meu tutorial em [6]. Recomendo usar o JACK [7] como sistema de som em vez do ALSA, pois foi dele que obtive os melhores resultados no que toca ao desempenho de entrada e saída (veja o **quadro 3: A respeito dos sistemas de som do Linux...**). Se depa-

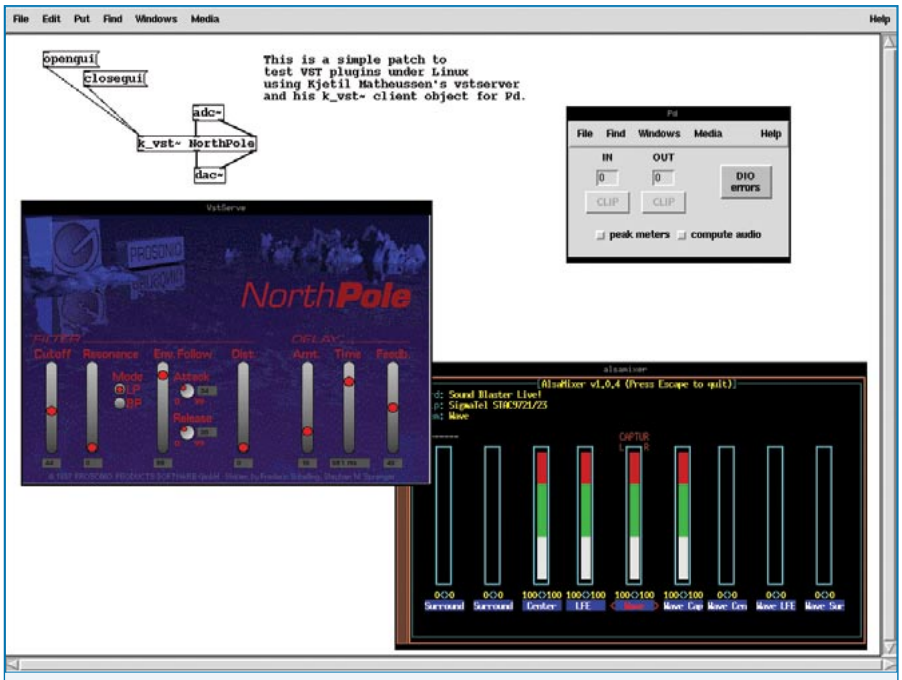

Figura 5: O objeto *k_vst~* do *Pd* hospedando um plugin VST.

Usando um plugin VSTi

Certo, certo, vamos acabar com o suspense. Como eu já havia dito, o sistema *vstserver* também inclui um cliente chamado *vsti*, que permite o uso de instrumentos VST (os chamados *plugins* VSTi). A sintaxe do cliente *vsti* é bem simples, veja a seguir:

```
# vsti /caminho/para/o/plugin_vsti.dll
```

Dê mais uma olhada na **figura 1**. Lá vemos o *vsti* rodando o sintetizador *Crystal*. O servidor *vstserver* está aberto e vemos o painel de conexões MIDI – o excelente *qjackctl* (qjackctl.sourceforge.net), de autoria de Rui Nuno Capela – ligando meu teclado MIDI externo ao *Crystal*. Além de conexões MIDI, o *qjackctl* também oferece um painel – não mostrado aqui – para interligar entidades de áudio.

rar com problemas como intermitências no som quando estiver usando o ALSA, experimente usar o JACK. Você nunca mais será o mesmo...

O cliente *ladspavst*

Usar um *plugin* LADSPA como hospedeiro para um *plugin* VST pode parecer bastante esquisito – afinal, são dois *plugins*, um dentro do outro, algo que lembra as *matryoshka* russas. O *ladspavst* de Kjetil realmente cria uma instância de *plugin* que age como hospedeiro de outros. O *ladspavst* em si é transparente para os usuários, já que apenas suas funções são importantes e não há necessidade de interface.

A **figura 6** mostra o *plugin* de filtro *NorthPole*, chamado do menu LADSPA em meu editor de som *Snd*, por sinal bastante "mexido". Como se pode ver na figura, os *plugins* VST estão listados como *VST plugin [plugin_name.dll] available from vstserver* (*plugin* VST disponível via *vstserver*). Não há diferença na forma de uso: os *plugins* VST são "plugados" em seu software da mesma forma que qualquer *plugin* LADSPA nativo. O *plugin ladspavst* permite dois modos de operação: os *plugins* VST podem adotar o visual do seu ambiente hospedeiro – no meu caso, Motif – ou podem mostrar o *plugin* em sua própria interface original, idêntica à que seria vista em um sistema Windows®.

Quadro 3: A respeito dos sistemas de som do Linux...

O sistema de som é a parte do kernel que cuida do hardware de áudio e sua interação com os programas. No Linux, o sistema de som padrão sempre foi o OSS/Free (o OSS aqui significa *Open Sound System* ou sistema de som aberto), mas recentemente o ALSA (*Advanced Linux Sound Architecture* – arquitetura de som avançada do Linux) foi adotado. O ALSA é hoje o sistema de som padrão do kernel das séries 2.5 e 2.6 em diante. É possível usar *plugins* LADSPA e VST no sistema antigo. Entretanto, essa prática é severamente desaconselhada: por favor, use o ALSA como base para seu sistema de áudio e MIDI. Você vai poupar bastante dores de cabeça.

O ALSA possui seu próprio sistema de *plugins*, que estendem sua paleta de recursos. Não é uma API genérica, mas serve para propósitos bastante úteis. Por exemplo, o *plugin dmix* permite que se faça mixagem por software em sistemas cujo hardware não reconhece fontes de áudio multiplexadas. Desse modo, sob condições normais, meu laptop pode rodar apenas um programa de áudio de cada vez, mas se o módulo `dmix` for "enxertado" no ALSA (mais precisamente no arquivo `~/.asoundrc`; consulte a documentação do ALSA para mais detalhes) é possível tocar um número impressionante de arquivos ao mesmo tempo.

Comunicando-se com o sistema de som no kernel, há uma outra entidade chamada *servidor de som*. Todos os aplicativos devem conversar com o servidor porque apenas ele tem permissão para "falar" com o kernel. Nessa categoria de software, uma das estrelas brilha mais que as outras.

O JACK [7] é um dos mais belos e impressionantes exemplos de software de áudio para Linux. Originalmente desenvolvido por Paul Davis (Ardour, Softwerk), o JACK evoluiu a ponto de se tornar *o* servidor de som do Linux, sem competidores. Robusto e de baixa latência, permite interconexão de I/O entre todos os clientes JACK enquanto oferece um mecanismo de controle de transporte para operação síncrona entre aplicativos compatíveis. Se suas necessidades de áudio tendem para o profissional, você *precisa* do JACK.

O `artsd` (*analog Real time syntheziser daemon*) e o `esd` (***Enlightened Sound Daemon***) são os servidores de som oficiais dos ambientes desktop KDE e GNOME, respectivamente. Foram desenvolvidos para propósitos relativamente simples, embora ambos possuam um potencial bastante interessante para eventos sonoros típicos de estações de trabalho (como aquele barulhinho chato quando você clica em alguma coisa ou alertas do seu cliente de mensagens instantâneas). O aRts inclui até uma API para criação de plugins. Entretanto, nenhum desses servidores foi desenvolvido pensando em um ambiente profissional de áudio e devem ser desabilitados caso queira usar o KDE ou o GNOME em seu estúdio de gravação com Linux.

O projeto FST

O sistema FST [8] (FreeST) é uma maneira alternativa de usar *plugins* VST e VSTi no Linux. Em vez de um servidor, implementou-se uma biblioteca. O desenvolvedor Paul Davis diz que o FST é "…uma solução mais apropriada para aplicações de áudio que pode trabalhar com muitos *plugins* VST, já que os chaveamentos de contexto do *vstserver* não trabalham muito bem quando diversos *plugins* VST são usados ao mesmo tempo". A desvantagem é que "…um *plugin* malcriado pode travar o hospedeiro, coisa que o *vstserver* evita".

O sistema inclui no momento a biblioteca `libfst` e o `jack_fst`, um utilitário de demonstração para carregar e rodar *plugins* VST/VSTi. Assim como no *vstserver*, compilar o FST requer as instalações completas do Wine e do JACK (incluindo código fonte) e os arquivos `AEffect.h` e `aeffectx.h`, do VST SDK da Steinberg.

As funções da biblioteca `libfst` podem ser compiladas em aplicativos como o gravador multipistas em disco rígido *Ardour*, o sistema de processamento e síntese de som *gAlan* e o seqüenciador MIDI *MuSE*. Os *plugins* VST ficam disponíveis para esses programas da mesma forma que os *plugins* nativos LADSPA.

O cliente *jack_fst* possui um "lançador" na linha de comando para rodar *plugins* VSTi como instrumentos independentes. O exemplo a seguir demonstra o processo com o sintetizador *Crystal*:

```
# jack_fst Crystal
```

A extensão do arquivo não é necessária, mas talvez seja preciso especificar o caminho completo para o diretório de *plugins*. O servidor JACK deve estar rodando e as conexões de áudio e MIDI necessárias já devem ter sido feitas pelo usuário – elas não são automáticas.

Problemas conhecidos

Tanto o *vstserver* quanto o *libfst* dependem fortemente de versões específicas do *Wine*. Como já dissemos, o site do *vstserver* oferece uma versão modificada do *Wine* que funciona a contento com o sistema. Entretanto, ele é baseado em código-fonte relativamente antigo; se por algum motivo você quiser uma versão mais nova do *Wine* pode se arriscar a perder a estabilidade do *vstserver*. O sistema FST é menos enjoado no tocante a versões do *Wine*, mas ainda assim falha em algumas delas. Consulte o tutorial sobre VST no Linux, disponível em [9], para saber as últimas notícias sobre a compatibilidade entre o *Wine* e o FreeST, além informações sobre outros problemas já detectados.

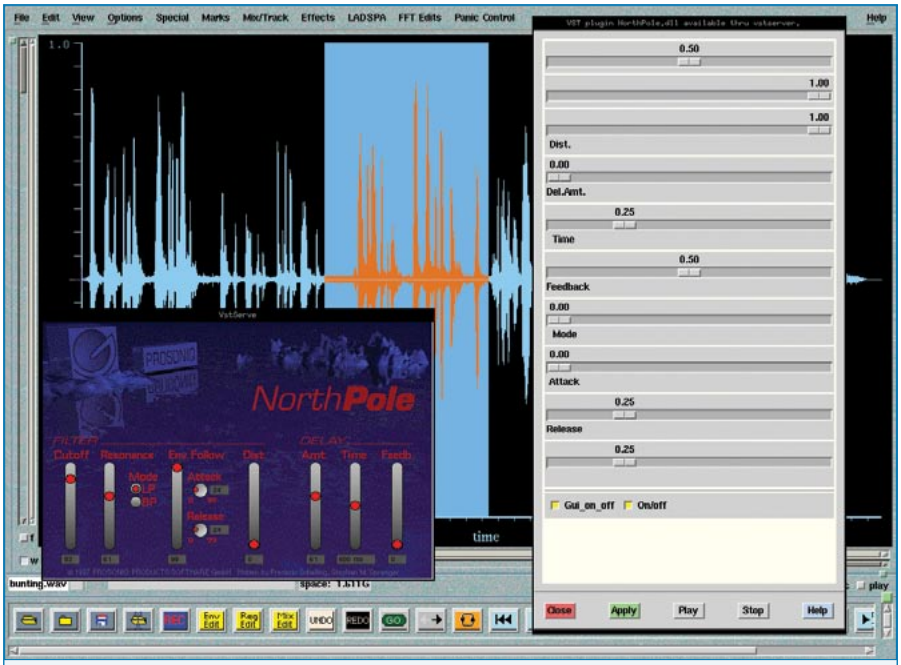

Figura 6: Um plugin VST em uso com o *Snd* via LADSPA.

Desligando os cabos

Esperamos que vocês tenham gostado desta breve descrição de como rodar *plugins* VST e VSTi no Linux. Os *plugins* de áudio se tornaram componentes essenciais em qualquer estúdio moderno de gravação e processamento de som e os usuários do Linux podem, agora, desfrutar tanto de seus excelentes *plugins* nativos no padrão LADSPA quanto do imenso acervo de *plugins* VST/VSTi antes disponíveis só para usuários do Microsoft Windows®. Trocando em miúdos, os músicos amigos do pingüim têm à sua disposição, numa só tacada, muito mais plugins que todos os outros músicos. ■

Informações

[1] LADSPA: http://www.ladspa.org
[2] DSSI: http://dssi.sourceforge.net
[3] *All Things LADSPA*: http://www.linux-sound.org/ladspa.html
[4] Site oficial do *vstserver*: http://www.notam02.no/arkiv/src/
[5] Pd - Pure Data: http://crca.ucsd.edu/~msp/software.html
[6] Usando plugins VST/VSTi no Linux: http://www.djcj.org/LAU/quicktoots/toots/vst-plugins/
[7] JACK: http://jackit.sourceforge.net/
[8] Site oficial do FST: http://linuxaudiosystems.com/fst/
[9] Compatibilidade entre Linux e plugins VST: http://www.djcj.org/LAU/ladspavst/
[10] Thac, repositório com pacotes multimídia para o Mandrake: http://rpm.nyvalls.se
[11] KVR Audio: http://www.kvraudio.com/
[12] Lista de discussão LAU (*Linux Audio Users*): http://www.linuxdj.com/audio/lad/subscribelau.php3
[13] Repositório de software de para edição de som e produção musical no Linux: http://linux-sound.org/
[14] Jack Audio Connection Kit - Qt Interface: http://qjackctl.sourceforge.net/
[15] Steinberg: http://www.steinberg.net/