

Uma olhada abrangente no Software Livre e afins

Projetos na incubadora

Os projetos de software livre abrangem uma vasta gama de utilidades. Talvez por isso mesmo seja difícil escolher a ferramenta perfeita. Nesta coluna tentamos colher os melhores frutos do cacho. Este mês demos uma espiadela no polêmico YaST2 para Debian, no aguardadíssimo KDE 4 e – para variar – no projeto Debian.

POR MARTIN LOSCHWITZ



A Novell, logo que adquiriu a SUSE, licenciou o YaST2 sob a licença GPL. O código fonte do programa já havia sido divulgado há bem mais tempo, mas a licença restringia o seu uso e distribuição – o que impedia seu uso e distribuição por terceiros. Essas mesmas restrições impossibilitaram, por muitos anos, que qualquer interessado pudesse baixar as imagens ISO dos CDs do SUSE Linux.

Yast4Debian

Representantes da Novell afirmaram que a empresa pretende fazer do YaST2 uma ferramenta universal de configuração, usável por qualquer distribuição. De acordo com a Novell, todas as distribuições têm o direito (e o dever) de modificar o código fonte do YaST para adaptá-lo a seus produtos. O primeiro grande projeto a meter os peitos na empreitada foi o Debian. Como é sabido de todos, a distribuição Debian não possui seu próprio painel de controle central, no qual os usuários podem configurar e administrar o sistema. Criar uma ferramenta abrangente como o YaST2 envolveria esforços descomunais, dignos de Hércules. O projeto *Yast4Debian* foi criado, então, para portar o YaST2 do SUSE 9.1 para o Debian. Espera-se para breve que o projeto atualize sua versão do YaST2 para deixá-lo em sincronia com o do SUSE 9.2.

O YaST2 engloba a biblioteca *Libyutil*, um núcleo básico e inúmeros módulos. O núcleo é apenas um interpretador para a linguagem de programação específica do YaST2, chamada

YCP. Ela combina vários elementos de outras linguagens mais comuns e é usada para desenvolver novos módulos. Tanto as bibliotecas do YaST2 como o núcleo são escritos em C++ e pré-configurados com *Autotools* para que possam ser compilados em qualquer sistema Linux.

O núcleo e a biblioteca são moleza. Osso duro mesmo vai ser portar cada um dos módulos. As interfaces com o usuário oferecidas pelo YaST2 – como a em modo texto, que usa a biblioteca *Ncurses*, e a gráfica, que usa a biblioteca *Qt* – podem ser compiladas em qualquer sistema Linux sem tropeços. Infelizmente, com os módulos a enxaqueca é digna de dispensa médica. Há, por exemplo, um que cuida apenas dos ícones do YaST2. Há tantas interdependências entre os módulos que algumas coisas não parecem fazer muito sentido à primeira vista. Por exemplo, para compilar o módulo de particionamento, é preciso já ter feito o mesmo com o *Autoyast* – que por sua vez tem muitas outras dependências, como o módulo de configuração de vídeo (servidor X11), que requer os módulos de mouse e teclado. Como se não bastasse, o *Autoyast* depende do módulo *YOU* – *Yast Online Update* e do módulo de configuração do gerenciador de *boot*.

“Tunando” o YaST2

A tarefa de modificar a infra-estrutura do programa – centrada no “jeito SUSE” de fazer as coisas – para outra distribuição é algo que vai muito além de simplesmente recompilar o YaST2

e seus módulos. No caso do Debian, o maior desafio é a instalação e administração de programas – o famoso *gerenciamento de pacotes*. Um módulo do YaST2 comunica-se com o banco de dados RPM sempre que necessário, mas o Debian usa seu próprio formato de pacotes. Por outro lado, outros módulos do YaST2 precisam de pouco ou nenhum retrabalho. Por exemplo, como tanto o Debian quanto o SUSE usam o sistema *shadow* de ocultação (“sombreamento”) de senhas, os desenvolvedores puderam usar o módulo de administração de usuários praticamente sem alterações.

Os desenvolvedores do projeto Yast4Debian já podem colher os primeiros frutos do suor de seus rostos: além do módulo de configuração de usuários (que já veio meio pronto), o particionador já funciona sem maiores problemas. Mas há ainda muito a fazer, e a adoção iminente da última versão do YaST2 (a que acompanha o SUSE 9.2) terá impacto bastante forte na carga de trabalho. Se depois de ler isto você estiver louco para pôs suas mãos na massa, dê uma passadinha pela página oficial do projeto [1] e fique por dentro do que e como fazer. Diversão garantida.

Qt 4 e o KDE 4

O ciclo de desenvolvimento do KDE forma, tradicionalmente, uma massa homogênea com o ciclo de desenvolvimento da biblioteca gráfica que o governa, a famosa Qt. Foi emblemático, portanto, quando o desenvolvimento do KDE 4.0 começou “na cola” do lançamento da versão 4.0 da Qt. O KDE 4.0 está sendo desenvolvido para tirar proveito dos novos recursos da biblioteca Qt e, ao mesmo tempo, eliminar algumas falhas de projeto que infestavam o KDE 3.

O intervalo entre os dois lançamentos faz parecer óbvio que muitas mudanças nos alicerces do KDE, provocadas por mudanças na Qt, tiveram que ser levadas a bom termo. A versão 4.0 da biblioteca gráfica foi lançada em 27 de junho – a versão 3 já tem quase quatro anos, lançada em 21 do outubro de 2001.

A maior mudança da Qt, entretanto, não é nos recursos, funcionalidades ou no código: é na licença. A biblioteca Qt originalmente possuía uma licença livre apenas para Linux. Com a versão 4, a Trolltech (empresa desenvolvedora da Qt) estendeu para o Windows a abrangência da licença GPL, mantendo ainda a já conhecida *licença comercial*. Alguns desenvolvedores já falam em portar o KDE para o Windows – já que os entraves legais foram dizimados, é apenas uma questão de tecnologia.

A Qt 4 também traz novidades para o usuário final. A biblioteca já pode trabalhar com arquiteturas de 64 bits, com mais desempenho nesses ambientes.

O novo “motor” gráfico, que recebeu o nome de *Arthur*, também merece destaque. Com ele, os usuários podem incluir fun-

ções de desenho e pintura em qualquer aplicativo Qt de forma simples. O Arthur é muito mais rápido que seu antecessor, o *QPainter* (da Qt 3) e “conversa” com mais desenvoltura com as interfaces gráficas de qualquer sistema operacional que use Qt. Isso significa que os desenvolvedores não precisam mais considerar as características específicas de cada interface com o usuário. O novo mecanismo de renderização de fontes tipográficas, o chamado *Scribe*, também acabou com os problemas visuais que afetavam a Qt 3.

Resumindo: a Qt 4 dá aos desenvolvedores do KDE uma base sólida e poderosa na qual construir a próxima geração de ambientes gráficos para o usuário. A página Wiki do projeto [2] já exhibe uma lista bastante ousada dos recursos sugeridos para o KDE 4. Entre outras coisas, os desenvolvedores planejam implementar um esquema de renderização de ícones e atualizações “visuais” de cada um dos componentes individuais do KDE. A usabilidade e conforto do usuário serão também aprimorados, já que todos os ícones e botões do sistema usarão os métodos de acesso padrão da Qt – o que contribui para uma maior padronização visual do ambiente.

O projeto Plasma [3] está estudando uma plástica geral na aparência do KDE 4. De acordo com os planos iniciais, vários elementos do KDE serão integrados, incluindo uma versão completamente reescrita do Centro de Controle, do *Kicker* (painel do KDE) e a introdução do *KDesktop*, que agirá como uma “janela raiz” na qual estarão contidas todas as outras janelas do KDE. No departamento de orgia visual, o *Superkaramba* – antes um componente opcional – permitirá que os programadores deliciem os olhos do usuário com hedonismos visuais em praticamente todos os aspectos do ambiente gráfico – tudo isso com um mínimo de esforço. Será possível, por exemplo, oferecer ao usuário uma barra de tarefas bastante parecida com a do Mac OS X. Com tudo isso, os desenvolvedores almejam chegar a um “projeto visual moderno e estonteantemente inovador”. Uma biblioteca de compatibilidade com o KDE 3 fará a ponte entre o KDE 4 e aplicativos desenvolvidos para seu antecessor.

Os programadores do KDE ainda não chegaram a um acordo quanto ao prazo para lançamento da nova versão do ambiente gráfico. Obviamente, não são só os voluntários que trabalham no KDE os únicos a suspirar pelo sucesso de seu desenvolvimento: há algumas semanas, a Trolltech contratou o programador Aaron Seigo, um dos grandes nomes do KDE. Para trabalhar na Qt? Não. Para que ele se concentre unicamente no desenvolvimento do KDE. A Trolltech parece realmente querer tornar o melhor aplicativo desenvolvido em Qt um objeto de grande prestígio. ➡

Festa dos Exterminadores

Na esteira do lançamento do Debian Sarge, a árvore de desenvolvimento já trabalha a todo vapor. Por exemplo, tudo já foi migrado para o GCC 4.0. Como era de se esperar, a incidência de insetos gosmentos cresceu para mais de 600. Sem outra alternativa, os desenvolvedores do Debian compraram vários tamancos pontudos (daqueles usados para esmagar barata no canto) e se puseram a perseguir os artrópodes melequentos.

Para acelerar o processo, Frank Lichtenheld organizou uma “festa dos exterminadores de insetos” no início de agosto. Uma festa dessas (chamada, em inglês, de *bug squashing party*) consiste de uma reunião de desenvolvedores em um canal de IRC para caçar *bugs* nos seus próprios pacotes e ajudar os outros membros da equipe a fazer o mesmo. Os administradores relaxam um pouco as rígidas regras de envio de patches para que todos possam participar. A dedetização grupal exterminou um total de 32 erros críticos e, com isso, muitos pacotes puderam passar da árvore *unstable* (instável) para a *testing* (em teste).

À época que os esforços de desenvolvimento do Sarge estavam em seus suspiros finais, festas dos exterminadores eram organizadas mais ou menos uma vez por semana, ajudando a nova versão a finalmente desencantar e ser considerada estável – e, até que enfim, ser lançada... No passado, festas como essa sempre deram um ótimo resultado e não seria diferente agora.

O futuro da Debian CD Tool

O lançamento do Sarge viu o catálogo de aplicativos disponíveis para o Debian crescer sobremaneira – são mais de 15.500 pacotes! Quem não tem uma conexão rápida com a Internet precisa conseguir os de alguma forma os CDs ou o DVD de instalação. Todos esses programas abarrotam um DVD de camada dupla ou 14 CDs. Com uma conexão à Internet, o usuário precisa só do primeiro CD para instalar o sistema; o resto pode ser instalado diretamente Internet.

Para que o primeiro CD faça seu trabalho como esperado, ele deve conter um conjunto completo de pacotes vitais para o sistema operacional. Além disso, quaisquer dependências necessárias têm que estar obrigatoriamente contidas nesse CD – referências a outras mídias são absolutamente proibidas. Uma ferramenta chamada *Debian CD Tool* [5] assegura que a mídia usada respeita essas condições, mas há dois detalhes que devem sempre ser levados em conta. Primeiro, o código fonte da ferramenta é monstruosamente ininteligível, já que inúmeros desenvolvedores o adaptaram ao longo dos anos às situações e condições sempre mutantes do Debian. Qualquer programador que queira usar o *CD Tool* para criar seus próprios conjuntos de discos especiais tem que enfiar a cara no código e tentar entender a reinante macarronada com bacalhau.

Há planos para mudar isso. Na última DebConf (*Debian Developer Conference*) em Helsinque, Finlândia, um grupo de programadores liderados por Steve McIntyre reuniram-se para discutir maneiras de simplificar a ferramenta. A reestruturação do código fonte é o ponto central; ao mesmo tempo, a equipe tentará atualizar a já obsoleta verificação de dependências. Steve McIntyre anunciou alguns dos objetivos do grupo na lista de discussão *Debian-Devel-Announce* [6].

Mas a ferramenta CD Tool encara problemas maiores que esse, alguns vindos de lugares inusitados. Para saber como distribuir os 15.500 pacotes nos 14 CDs da distribuição, a ferramenta baseia suas decisões nos dados da Pesquisa de Popularidade Debian (*Debian Popularity Contest*). Pense nela como “a parada de sucesso Debian”. Os usuário da distribuição podem instalar em seus sistemas um pacote chamado *popularity-contest*. Esse programa envia a servidores específicos para esse fim uma lista dos pacotes instalados na máquina, uma vez por semana. As estatísticas produzidas a partir desses dados são avaliadas para decidir a ordem de prioridade dos pacotes nos CDs do Debian (quanto maior a prioridade, mais próximo do primeiro CD o pacote fica).

Infelizmente, o número de usuários que transmitem as listas de pacotes está escasseando. Isso levanta questões a respeito da confiabilidade da lista. Em uma mensagem à lista Debian-Devel [7], Petter Reinholdtsen alertou para a iminente extinção desses usuários e, por extensão, da lista de pacotes. Sem essa lista, o Debian CD Tool é nada mais que inútil.

Isso é tu-tu-tudo, pe-pessoal...

...ao menos por este mês, mas tenho um último pedido antes de partir: se você quiser recomendar um programa que queira ver apresentado em *Projetos na Incubadora*, por que não me mandar um email (em inglês ou alemão) com sua sugestão [8]? Aguardo seus comentários! ■

INFORMAÇÕES

- | | |
|-----|---|
| [1] | YaST4Debian: yast4debian.alioth.debian.org |
| [2] | Objetivos do KDE 4: wiki.kde.org/KDE+4+Goals |
| [3] | Projeto Plasma: plasma.kde.org |
| [4] | Bluetooth Alsa: bluetooth-alsa.sourceforge.net |
| [5] | Debian CD Tool: www.debian.org/CD |
| [6] | Steve McIntyre e o Debian CD Tool: lists.debian.org/debian-devel-announce/2005/07/msg00005.html |
| [7] | Peter Reinholdtsen e a popularidade dos pacotes: lists.debian.org/debian-devel/2005/07/msg01269.html |
| [8] | Dicas e sugestões (mensagens em inglês ou alemão): projects@linux-magazine.com |