

LINUX MAGAZINE
NÚMERO 9
JUNHO 2005
PCs VIRTUAIS WINE VMWARE UML MAC-ON-LINUX PEARPC XEN BLUEFISH GENTOO APT SPARC KNODA VPN GDBM



TRUQUES RÁPIDOS
Dicas para ganhar tempo



PAPO DE BOTEQUIM:
"Strings", variáveis e expansão de parâmetros

TECNOLOGIA SEM LIMITES

LINUX
MAGAZINE

LINUX

MAGAZINE

exemplar de assinante
venda proibida



gentoo 2005.0
A DISTRIBUIÇÃO MAIS
RÁPIDA DO PLANETA!



PCs VIRTUAIS

Use programas do Windows no Linux
VMware 4.5.2: dois PCs em uma só máquina
User-mode Linux: um Linux dentro do outro
Rode o Mac OS X dentro do Linux
Xen: virtualização e alto desempenho

Sempre ligado
Conexões VPN com alta disponibilidade

Soltando faíscas
Rode o Linux em máquinas SPARC

Veja também:
O editor de HTML Bluefish
knoda, o banco de dados do KDE
APT em distribuições baseadas em RPM
Programar bancos de dados com GDBM
Aprenda a instalar o Gentoo Linux



R\$14,90
€ 6,50
Ed. 09
06/2005

00009

9 771806 942009

EXPEDIENTE EDITORIAL

Editores

Rafael Peregrino da Silva, rperegrino@linuxmagazine.com.br
Rafael Pereira Rigues, rrrigues@linuxmagazine.com.br

Direção de Arte e Projeto Gráfico

Luciano Hagge Dias, lhagge@linuxmagazine.com.br

Centros de Competência

Centro de Competência em Software:
Oliver Frommel, ofrommel@linux-magazine.com
Centro de Competência em Hardware:
Mirko Dölle, mdoelle@linux-magazine.com
Centro de Competência em Redes e Segurança:
Achim Leitner, aleitner@linux-magazine.com

Correspondentes & Colaboradores

Alexandre Barbosa, Andrea Müller, Andreas Grytz, Armijn Hemel, Augusto Campos, Bernd Villiger, Christiano Anderson, Fabrizio Ciachi, Hagen Höpfner, Johannes Hubertz, Jörg Luther, Julio Cezar Neves, Lucas Brasilino, Marcel Hilzinger, Marcelo Veriato Lima, Marco Antônio Bocardo Sinhoreli, Martin Loschwitz, Mirko Albrecht, Oliver Frommel, Thomas Hümmler, Willian Ferraz, Zack Brown.

Confeção do CD-ROM

Ricardo D. Dalceno, rdalceno@linuxmagazine.com.br

Tradução e Revisão

Henrique Cesar Ulbrich, henrique.cesar@ermida.com
Julia Vidili, julia.vidili@ermida.com

Design da Capa

Pinball, info@pinball-werbeagentur.de

Anúncios:

www.linuxmagazine.com.br/Advertise

Brasil

Wladimir Porto, anuncios@linuxmagazine.com.br
Tel./Fax: +55 (0)11 2161 5400
Cel.: +55 (0)11 8457 1761

Todos os países (exceto Alemanha, Áustria e Suíça)

Brian Osborn, ads@linux-magazine.com
Tel.: +49 (0)6509 910 495
Fax: +49 (0)6509 910 497

Alemanha, Áustria e Suíça

Osmund Schmidt, anzeigen@linux-magazine.com
Tel.: +49 (0)6335 9110
Fax: +49 (0)6335 7779

Diretoria

Rafael Peregrino da Silva, rperegrino@linuxmagazine.com.br
Claudio Bazzoli, cbazzoli@linuxmagazine.com.br

Linux Magazine

Av. Eng.º Luis Carlos Berrini, 1500
Conj. 103 – Brooklin Novo
04571-000 – São Paulo – SP – Brasil
Tel.: +55 (0)11 2161 5400
Fax: +55 (0)11 2161 5410

Assinaturas:

www.linuxmagazine.com.br/Subs
Preço: (12 edições incluindo CD mensal): R\$143,00
Email: assinaturas@linuxmagazine.com.br
Preço Unitário: R\$14,90

Na Internet:

www.linuxmagazine.com.br – Brasil
www.linux-magazin.de – Alemanha
www.linux-magazine.com – Portal Mundial
www.linuxmagazine.com.au – Austrália
www.linux-magazine.ca – Canadá
www.linux-magazine.es – Espanha
www.linux-magazine.pl – Polônia
www.linux-magazine.co.uk – Reino Unido
www.linux-magazin.ro – Romênia

Apesar de todos os cuidados possíveis terem sido tomados durante a produção desta revista, a editora não é responsável por eventuais imprecisões nela contidas ou por consequências que advêm de seu uso. A utilização de qualquer material da revista bem como do CD-ROM incluso ocorre por conta e risco do leitor. O CD-ROM foi testado extensivamente e, até onde pudemos verificar, se encontra livre de qualquer vírus ou outro tipo de software de conteúdo malicioso, bem como de defeitos.

Nenhum material pode ser reproduzido em qualquer meio, em parte ou no todo, sem permissão expressa da editora. Assume-se que qualquer correspondência recebida, tal como cartas, emails, faxes, fotografias, artigos e desenhos, são fornecidos para publicação ou licenciamento a terceiros de forma mundial não exclusiva pela Linux New Media do Brasil, a menos que explicitamente indicado.

Linux é uma marca registrada de Linus Torvalds.

Linux Magazine é publicada mensalmente por: Linux New Media do Brasil Editora Ltda., São Paulo/SP, Brasil.

Distribuição: Fernando Chinaglia Distribuidora.

Direitos Autorais e Marcas Registradas © 2005:
Linux New Media do Brasil Editora Ltda.

Impressão e Acabamento: PROL Gráfica

ISSN 1806-9428



em processo de filiação

Impresso no Brasil

Oportunidades desperdiçadas...

Prezado leitor, prezada leitora da Linux Magazine,

observando a oferta de Software Livre em alguns segmentos do mercado, constatamos que uma oportunidade vem sendo desperdiçada: a de atender a uma demanda para sistemas dedicados usando Linux. Consideremos os cinco segmentos a seguir:

- ⇒ Pontos de venda (PDV) e automação comercial,
- ⇒ Controle de acesso e de vigilância em edifícios,
- ⇒ Cobrança em estacionamentos,
- ⇒ Telemarketing/call center,
- ⇒ Gestão Empresarial e CRM.



Já existem soluções para pontos de venda rodando em Linux. A maioria deles usa tecnologia proprietária, como é o caso do Calypso, da Unisys, utilizado pelo Carrefour; ou do sistema utilizado pelo Grupo Pão de Açúcar, desenvolvido em parceria com a Itautec. Há também o SACI (www.eacnet.com.br/saci_mysql), da EAC Software, um pacote integrado de automação comercial. O único sistema de código aberto que encontramos aqui no Brasil em nossa singela análise foi o Stoq (www.stoq.com.br), da Async Open Source, cuja última atualização data de maio de 2002.

Aparatos para controle de acesso a edifícios e de vigilância poderiam ser facilmente desenvolvidos com base no projeto ZoneMinder (www.zoneminder.com). Ele permite a captura, análise, gravação e monitoramento de vídeo proveniente de uma ou mais câmeras, que podem estar em rede, conectadas a um computador com Linux.

Para a cobrança em estacionamentos, a aplicação seria um misto de PDV e central de vigilância, em conjunto com um dispositivo de controle de cancela. Como se vê, nada que exija um estágio na NASA para ser implementado.

Temos ainda o nicho de telemarketing para call centers. Aqui no Brasil, a G4 Solutions está aprontando o G!Neration, para rodar também no Linux – atualmente, o aplicativo está disponível apenas para Windows®. Uma alternativa livre é o vtiger (www.vtiger.com) que, apesar de ser de uso genérico, serviria muito bem como base para a criação de um controle de telemarketing profissional.

Por fim temos os sistemas de gestão empresarial e CRM. Há diversas opções livres no mercado: Compiere, SQL-Ledger, Fistera e GNU Enterprise, entre muitas outras. Algumas delas já se encontram em uso em linhas de produção bastante complexas. É o caso do Compiere, usado pela CCE aqui no Brasil e pela Goodyear no exterior.

Sistemas como os citados acima, rodando em plataforma livre, contribuem fortemente para a diminuição de custos de investimento e manutenção. A ausência de gastos com licenças, o fato de não haver necessidade de software adicional (como antivírus) e os longos tempos de operação dessas soluções são, certamente, atrativos bastante sedutores para o cliente. Além disso, contribuem para afastar o consumidor da pirataria, infelizmente ainda muito freqüente em nosso país. A demanda para tais soluções baseadas em Linux existe; o sistema é robusto e todas as ferramentas de desenvolvimento são disponíveis livremente. A base para criar tais sistemas também está aí. Quem investir de maneira profissional no desenvolvimento de tais tecnologias para Linux terá, com certeza, grandes vantagens para apresentar e vai se diferenciar da concorrência.



Rafael Peregrino da Silva
Editor

Permissão de escrita

06

Notícias

Dicas de [In]segurança

O Pingüim ganha espaço

Pesquisa da Fundação Getúlio Vargas mostra crescimento do Linux entre as empresas.

Notícias do Kernel

Pingüim de casaca azul

Conversamos com Haroldo Hoffmann, coordenador de ações da IBM no Brasil, e Cezar Taurion, gerente de novas tecnologias.

Mundo livre em revista

08

08

11

12

14

16

Capa

Hóspede & anfitrião

Água ou vinho

Se você precisa usar um aplicativo Windows® no Linux, não há opção melhor que o *Wine*.

PC numa caixa

Analisamos o *VMware 4.5.2*.

Linux no Linux

Acredite, é possível rodar o Linux... dentro do Linux!

Uma maçã por dia...

Veja as várias formas de rodar o Mac OS X no Linux.

Multiplicando pingüins

Xen, um software de virtualização rápido e eficiente.

23

23

24

28

32

36

40

Análises

Peixes na rede

Conheça o *Bluefish*, um versátil editor de HTML.

46

46

Tutoriais

Hummm... casa arrumada, hein?

Veja como criar formulários no *knoda*.

Gentoo Linux

Saiba mais sobre esta versátil distribuição Linux.

Misturaça de pacotes

Aprenda a usar o *APT* em uma distribuição baseada no sistema de pacotes *RPM*.

Soltando faíscas

Cuidados ao rodar o Linux em máquinas *SPARC*.

48

48

52

60

65

23 Hóspede & Anfitrião

Por que não rodar aplicativos do Windows® ou o próprio Windows® no Linux? Por que não rodar o Mac OS X ou o Linux, dentro de um sistema Linux “hospedeiro”? Os desenvolvedores gostam dos sistemas virtuais porque um aplicativo pode derrubar o PC virtual sem derrubar o sistema real que o abriga. Os administradores de sistemas usam máquinas virtuais para testar versões do kernel ou experimentar novas configurações de rede. Os usuários querem apenas rodar seu aplicativo proprietário favorito. Neste mês, mostraremos o caminho para a computação virtual no Linux, abordando o popular *Wine*, o tradicional *VMware*, os sistemas de virtualização *User-mode Linux* e *xen* e softwares como o *PearPC* e *Mac-on-Linux*, que permitem a até pouco tempo inacreditável façanha de rodar o Mac OS X dentro do Linux.

46 Peixes na rede

A construção de projetos web complexos com suporte a CGI pode ser algo muito confuso. Um editor HTML como o *Bluefish*, com suporte a gerenciamento de projeto e às mais populares linguagens de script ajuda você a se manter na linha. A versão atual, 1.01, oferece aos usuários alguns interessantes recursos novos.



48 **Gentoo Linux**

O Gentoo Linux nada mais é do que uma distribuição Linux voltada para a otimização e personalização de pacotes. Seu alto desempenho, configurabilidade e uma enorme equipe de desenvolvedores espalhados pelo mundo fizeram desta uma das distribuições mais populares.

56 **Misturanc a de pacotes**

O *Advanced Packaging Tool* (APT), origin rio da distribui o Debian,   o sistema mais avan ado (e simples de usar) para gerenciamento e instala o de programas no Linux. Mas e se voc  n o usar o Debian? Este artigo descreve como usufruir do poderoso sistema APT em outras distribui es baseadas no sistema de pacotes RPM, como o SUSE LINUX

69 **Seguran a de ponta a ponta**

Navegando em um cybercaf , acessando a rede a partir de uma conex o p blica ou compartilhando a banda larga na casa de um amigo? Se voc  n o tem confian a nos caminhos pelos quais seus dados trafegam, criar um t nel SSH pode ser uma solu o.

70 **Caminhos alternativos**

O IPsec impede que muitos dos truques oferecidos pelos produtos de alta disponibilidade funcionem. Mostramos um caminho alternativo.

75 **Programando com GDBM**

O GDBM, ou *GNU Database Manager*,   uma biblioteca escrita na linguagem C que possibilita ao programador a manipula o de informa es em arquivos de banco de dados. Deixe que ele gerencie os dados de seus aplicativos e tenha menos dores de cabe a.

80 **Truques r pidos**

Softwares s o como pessoas: h  detalhes e segredinhos que voc  s o descobre depois de anos de relacionamento. Conhe a aqui algumas dicas para softwares populares no mundo Linux que podem facilitar sua vida e poupar tempo. Veja aqui as cartas que o *OpenOffice.org*, *KDE*, *Evolution* e at  mesmo o bom e velho terminal escondem em suas mangas.



69	SysAdmin
69	Seguran�a de ponta a ponta Se voc� n�o tem confian�a na rede pela qual seus dados trafegam, os t�neis SSH podem ser uma solu�o.
70	Caminhos alternativos O IPsec impede que muitos dos truques oferecidos pelos produtos de alta disponibilidade funcionem. Mostramos um caminho alternativo �s conex�es IPsec.
75	Programa�o
75	Programando com GDBM Deixe que o GNU Database Manager gerencie os dados de seus aplicativos e tenha menos dores de cabe�a.
79	Linux User
80	Truques r�pidos Conhe�a algumas dicas para softwares populares no mundo Linux que podem facilitar sua vida.
84	Papo de botequim – parte IX Vamos aprender mais sobre formata�o de cadeias de caracteres, conhecer as principais vari�veis do Shell e nos aventurar no mundo da expans�o de par�metros.
90	Comunidade
90	Planeta GNU Saiba mais sobre os perigos da Propriedade Intelectual, e aprenda que adotar o Software Livre n�o significa ter de abrir m�o de tecnologias populares.
92	Projetos na incubadora YavST2 no Debian, <i>Battle for Wesnoth</i> e muito mais.
96	Servi�os
96	Eventos / Mercado Linux
97	Quadrinhos / Anunciantes
98	Na pr�xima edi�o

Cartas para o editor

Permissão de escrita

Ubuntu

» Gostaria de fazer um “adendo” à matéria sobre o Ubuntu publicada na sétima edição da Linux Magazine: no subtítulo *A Reconciliação da Comunidade* o autor diz que, se um determinado programa não possui atalho na área de trabalho, a única solução é chamá-lo pelo terminal. Na verdade, se você quiser, por exemplo, incluir um ícone no menu *Acessórios* (ou qualquer outro), basta ir até lá, clicar com o botão direito sobre o submenu e descer até a opção *Entire menu | Add new item to this menu*. Preencha os campos *Name* e *Command* (com o caminho completo até o executável) e escolha um ícone clicando sobre o botão *No Icon*. Pronto!

O artigo também não explicou porque o LiveCD do Ubuntu traz programas para o Windows, mas eu acho que é porque não tem explicação mesmo...

Gostaria ainda de dizer que existem um fórum (ubuntu.linuxval.org) e uma lista de discussão (lists.ubuntu.com/mailman/listinfo/ubuntu-pt), ambos em português, onde brasileiros, portugueses, moçambicanos, angolanos etc., em meio a muitos “ecrãs”, “ratos” e “utilizadores”, esforçam-se para ajudar a todos que têm qualquer tipo de dúvida ou problema com a distribuição.

Para terminar: como já estou usando o Ubuntu 5.04, Hoary Hedgehog, os discos que recebi ficaram sem utilidade para mim. Resolvi, então, despachá-los para diversos amigos, acompanhados de uma cópia das duas matérias da LM. Acho que fiz uma coisa boa: agora, os felizardos têm a possibilidade de conhecer uma excelente distri-

buição, além de sentir o gostinho de uma revista que honra o mundo do software livre. Esse é o espírito do Ubuntu.

José Tadeu de Barros

Juiz de Fora – MG

Obrigado pela dica sobre como adicionar atalhos aos menus e pelos endereços dos pontos de encontro da comunidade. Quanto aos programas para Windows, a lógica é simples: eles estão no CD para que os usuários possam experimentar as principais “estrelas” do Software Livre sem ter de instalar uma distribuição Linux e sem ficar presos às limitações de um LiveCD (como a incapacidade de salvar arquivos).

Você fez uma coisa boa ao apresentar seus amigos ao Software Livre, ainda mais com uma distribuição “bem-arrumada” como o Ubuntu, que vai causar uma ótima impressão. Lembre-se de ficar por perto para ajudá-los com eventuais dúvidas. Com certeza, mais tarde eles irão lhe agradecer.

Informix

» Sou usuário de Software Livre, mas não tenho nenhum preconceito sobre a origem do programa. Pode ser livre ou não, se ele atende às minhas expectativas, principalmente de custo, aprendizagem e qualidade, eu uso. Achei estranha uma notícia publicada na página 17 da sétima edição sobre nova versão do Informix, da IBM.

O Gerenciador de Banco de Dados Informix é proprietário e acho que nunca existiu na revista uma notícia positiva sobre programas proprietários. E essa notícia está, justamente, elogiando um deles. Engraçado é que na mesma página há uma nota negativa sobre a Microsoft.

Já vi muitas pessoas do movimento de Software Livre encherem a IBM de elogios porque ela abriu um pouco de seu código, apóia, dá suporte e usa o Linux. É claro que todos nós sabemos que isso ocorre porque eles querem lucro. Ninguém se lembra dos órfãos do OS2? Em breve teremos os órfãos do AIX. Não faz sentido manter um sistema operacional proprietário se tem um livre bem melhor. O OS2 não dava lucro e foi abandonado. [...]



Creio que a notícia está fora da linha editorial da revista. Daqui uns dias vamos ter notícias positivas sobre a Microsoft...

Cleber Vieira Baptista

Estranho, você diz não discriminar o software por sua origem ou modelo de licenciamento, mas seu comentário deixa transparecer um sentimento exatamente oposto. Assim como o Oracle, o Informix é um SGBD popular e roda sobre a plataforma Linux, portanto consideramos a notícia pertinente e de interesse para nossos leitores, muitos dos quais são DB Admins (e provavelmente há entre eles usuários do Informix).

Não há nada errado no fato de uma empresa buscar o lucro (afinal, é para isso que elas existem). O fato é que as contribuições da IBM ao Linux e ao Software Livre, independente das intenções da empresa, são inestimáveis e “eternas”, pelos termos sob os quais o código foi licenciado, e devemos aplaudi-la por isso. Graças ao “interesse” da empresa em adaptar o sistema para suas máquinas de grande porte como uma alternativa ao AIX, foram feitos vários avanços nas áreas de multiprocessamento, suporte a sistemas NUMA, sistemas de arquivos com Journaling (JFS, que veio do OS/2) e suporte a novas arquiteturas e processadores, como a POWER4 (usada em Mainframes) e PPC970 (o famoso “PowerPC G5”, processador das novas máquinas da Apple). Isso sem falar no Eclipse, contribuições ao projeto Apache. A lista é longa...

O povo quer debater

» Gostaria de sugerir que a revista criasse um fórum no site para que os leitores possam discutir e trocar informações a respeito das edições e seções da revista. A estrutura poderia ser algo como um tópico por edição, no qual seriam discutidos os artigos e temas da edição daquele mês. Um moderador verificaria se o conteúdo é pertinente ao tópico, como em qualquer fórum.

Acho que, com esse fórum, os leitores poderiam aproveitar melhor o conteúdo da revista e a equipe da Linux Magazine teria como avaliar se o conteúdo foi bem aceito, quais as principais dúvidas dos usuários, etc.

Nelson R. Estevam Junior

Sugestão anotada, Nelson. Estamos planejando uma série de mudanças em nosso site para um futuro próximo, e a criação de fóruns de discussão fazem parte da lista de possíveis melhorias.

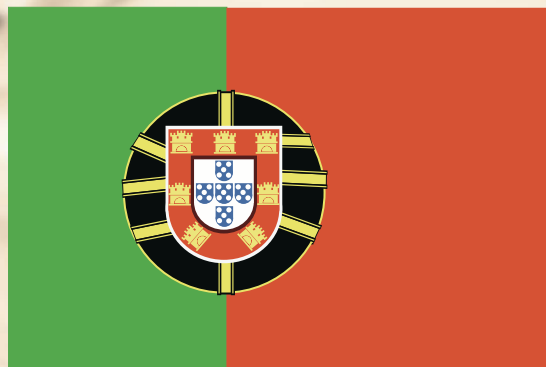
Eu tenho a força!

» Vários leitores nos escreveram pedindo a senha de root do Ubuntu 4.10, distribuído em nossa sétima edição. Por motivo de segurança, no Ubuntu o usuário root vem “desabilitado” e todas as operações nas quais poderes de root são necessários são feitas com o utilitário `sudo`. Se você precisar trabalhar como root durante um tempo, pode iniciar uma sessão com o comando `sudo su`. Você permanecerá como root até encerrar a sessão com o comando `exit`.

Se você realmente precisa habilitar o usuário root, se sabe o que está fazendo e tem a supervisão de um adulto responsável com mais de 80 anos, use o comando `sudo passwd`. Digite uma senha para o root, confirme-a e pronto. Agora você pode iniciar uma sessão do usuário todo-poderoso com o login root e a senha que definiu.

Saudações de além-mar

» Li do princípio ao fim a 1ª Edição da 'Linux Magazine', que muito apreciei e que espero continue a publicar-se para bem de toda a comunidade amante do Software Livre de todo o mundo. Gostei sobretudo de vossa notícia sobre o evento comemorativo do lançamento do primeiro número da revista.



Exprimo os meus votos de que continuem a trabalhar pelo desenvolvimento dos projectos que figuram nas páginas do jornal e a merecer a máxima confiança depositada no saber que partilham com todos os leitores.

Antonio Manuel da Silva Contente Lisboa, Portugal

Antonio, agradecemos os elogios. Ficamos contentes em saber que a revista está sendo bem-recebida em Portugal. Aguarde as próximas edições!

Rato de Biblioteca

» Gostaria de ver na revista artigos sobre software para gerenciamento de bibliotecas, como o PHL 8.0 (Personal Home Library) e o GNUteca, ambos nacionais.

Alexandre Bruno Campo Grande – MS

ESCREVA PRA GENTE

Se você tem dúvidas sobre o mundo Linux, críticas ou sugestões que possam ajudar a melhorar nossa revista, escreva para cartas@linuxmagazine.com.br. Devido ao

volume de correspondência, é impossível responder a todas as mensagens, mas garantimos que elas são lidas e analisadas. As mais interessantes são publicadas nesta seção. Para dúvidas ou críticas referentes à sua

assinatura da Linux Magazine, use o endereço: assinaturas@linuxmagazine.com.br



Dicas de [In]segurança

❑ Emacs

O Emacs é um editor de texto poderoso, altamente personalizável, auto-documentador e independente de arquitetura.

Max Vozeler descobriu diversas vulnerabilidades de formatação de cadeias de caracteres (*strings*) no utilitário *move-mail* do Emacs. Se um usuário se conectasse a um servidor POP malicioso, um agressor poderia executar código arbitrário com as permissões do usuário

que estivesse rodando o Emacs. O projeto *Common Vulnerabilities and Exposures* (cve.mitre.org) deu a essa falha o código CAN-2005-0100.

Todos os usuários do Emacs são aconselhados a atualizar o programa para a versão mais nova. ■

Referência no Gentoo: GLSA 200502-20 / Emacs

Referência no Mandrake: MDKSA-2005:038

Referência no Red Hat: RHSA-2005:110-06

Referência no SUSE: SUSE-SR:2005:006

❑ Squid

O Squid é um proxy para a web desenvolvido para sistemas Unix.

Um estouro de buffer foi encontrado no interpretador *Gopher*. Essa falha permite que um servidor Gopher remoto possa derrubar um proxy Squid que leia dados a partir dele. Embora os servidores Gopher sejam bastante raros hoje em dia, uma página web maliciosa (por exemplo) poderia redirecionar o usuário ao site

Postura das principais distribuições Linux quanto à segurança

Distribuição	Referência de Segurança	Comentários
Conectiva	Info: http://distro2.conectiva.com.br/ Lista: seguranca-admin@distro.conectiva.com.br e http://distro2.conectiva.com.br/lista/ Referência: CLSA-... ¹	Possui uma página específica; não há link para ela na página principal. Os alertas são sobre segurança, mas distribuídos através de emails assinados com a chave PGP da empresa para assegurar sua autenticidade. Contém também links para os pacotes atualizados e para fontes de referência sobre o problema sendo corrigido.
Debian	Info: http://www.debian.org/security/ Lista: http://lists.debian.org/debian-security-announce/ Referência: DSA-... ¹	Alertas de segurança recentes são colocados na homepage e distribuídos como arquivos HTML com links para os patches. O anúncio também contém uma referência à lista de discussão.
Gentoo	Info: http://www.gentoo.org/security/en/gsla/index.html Fórum: http://forums.gentoo.org/ Lista: http://www.gentoo.org/main/en/lists.xml Referência: GLSA: ... ¹	Os alertas de segurança são listados no site de segurança da distribuição, com link na homepage. São distribuídos como páginas HTML e mostram os comandos necessários para baixar versões corrigidas dos softwares afetados.
Mandrake	Info: http://www.mandrakesecure.net Lista: http://www.mandrakesecure.net/en/mlist.php Referência: MDKSA-... ¹	A MandrakeSoft tem seu próprio site sobre segurança. Entre outras coisas, inclui alertas e referência a listas de discussão. Os alertas são arquivos HTML, mas não há links para os patches.
Red Hat	Info: http://www.redhat.com/errata/ Lista: http://www.redhat.com/mailling-lists/ Referência: RHSA-... ¹	A Red Hat classifica os alertas de segurança como "Erratas". Problemas com cada versão do Red Hat Linux são agrupados. Os alertas são distribuídos na forma de páginas HTML com links para os patches.
Slackware	Info: http://www.slackware.com/security/ Lista: http://www.slackware.com/lists/ (slackware-security) Referência: [slackware-security] ... ¹	A página principal contém links para os arquivos da lista de discussão sobre segurança. Nenhuma informação adicional sobre segurança no Slackware está disponível.
SUSE	Info: http://www.novell.com/linux/security/ Lista: http://www.novell.com/linux/download/updates/ Referência: suse-security-announce Referência: SUSE-SA ... ¹	Após mudanças no site, não há mais um link para a página sobre segurança, que contém informações sobre a lista de discussão e os alertas. Patches de segurança para cada versão do SUSE LINUX são mostrados em vermelho na página de atualizações. Uma curta descrição da vulnerabilidade corrigida pelo patch é fornecida.

¹ Todas as distribuições indicam, no assunto da mensagem, que o tema é segurança.

Gopher malicioso, ou conter um quadro (*frame*) apontando para esse site. O projeto *Common Vulnerabilities and Exposures* (cve.mitre.org) deu a essa falha o código [CAN-2005-0094](#).

Uma falha de estouro de inteiros foi encontrada no interpretador de mensagens WCCP. É possível derrubar o Squid se o agressor for capaz de enviar uma mensagem WCCP mal-formada com um endereço de origem “falsificado” (*spoofed*) que remeta ao *home router* do Squid. O projeto *Common Vulnerabilities and Exposures* (cve.mitre.org) deu a essa falha o código [CAN-2005-0095](#).

Também no interpretador WCCP foi encontrado um estouro de buffer. O resultado é o usual: um agressor pode mandar pacotes malformados que tirariam o Squid do ar ou, o que é pior, permitiriam executar código arbitrário. O projeto *Common Vulnerabilities and Exposures* (cve.mitre.org) deu a essa falha o código [CAN-2005-0211](#).

Um vazamento de memória foi encontrado no módulo de autenticação *NTLM fakeauth_auth*. Um agressor poderia colocar o servidor Squid sob uma carga pesadíssima de tráfego, obrigando o módulo a consumir uma quantidade fabulosa de memória, resultando numa negação de serviço (*Denial of Service*, ou DoS). O projeto *Common Vulnerabilities and Exposures* (cve.mitre.org) deu a essa falha o código [CAN-2005-0096](#).

Uma falha de derreferência a um ponteiro nulo (*NULL pointer*) foi encontrada também no módulo *NTLM fakeauth_auth*. O projeto *Common Vulnerabilities and Exposures* (cve.mitre.org) deu a essa falha o código [CAN-2005-0097](#).

Foi encontrada uma falha de validação de usuário no módulo *squid_ldap_auth*. A falha permite que se “estufe” o nome do usuário com espaços em branco. Com isso, é possível contornar regras explícitas de controle de acesso ou tirar proveito de uma administração de usuários confusa.

O projeto *Common Vulnerabilities and Exposures* (cve.mitre.org) deu a essa falha o código [CAN-2005-0173](#).

A maneira como o Squid trata as respostas a requisições HTTP precisa de retrabalho. Como está hoje, um servidor web malicioso poderia enviar uma série de respostas HTTP especialmente manipuladas para “envenenar” o cache do Squid. Com o cache envenenado, os usuários podem ser remetidos a páginas diferentes das que eles pediram – e isso pode ser usado em benefício do agressor. O projeto *Common Vulnerabilities and Exposures* (cve.mitre.org) deu a essa falha dois códigos: [CAN-2005-0174](#) e [CAN-2005-0175](#).

Em outra falha semelhante, o Squid não consegue tratar corretamente cabeçalhos HTTP muito grandes em respostas a requisições. Um servidor web malicioso poderia enviar um cabeçalho especialmente criado para – novamente – envenenar o cache do Squid. O resultado é o mesmo: com o cache envenenado, os usuários podem ser remetidos a páginas diferentes das que pediram. O projeto *Common Vulnerabilities and Exposures* (cve.mitre.org) deu a essa falha o código [CAN-2005-0241](#). ■

Referência no Debian: [DSA-688-1 squid](#)

Referência no Gentoo: [GLSA 200502-25 / Squid](#)

Referência no Mandrake: [MDKSA-2005:047](#)

Referência no Red Hat: [RHSA-2005:060-20](#)

Referência no SUSE: [SUSE-SA:2005:008](#)

❑ Firefox

O *Mozilla Firefox* é um navegador de Internet de código aberto e livre.

Foi encontrada uma falha nas funções de manipulação de cadeias de caracteres (ou *strings*) do Firefox. Se um site malicioso puder usar isto para exaurir a memória na máquina do cliente, seria possível executar código arbitrário. O projeto *Common Vulnerabilities and Exposures* (cve.mitre.org) deu a essa falha o código [CAN-2005-0255](#). ➡

Outra falha foi encontrada, desta vez no tratamento de janelas *pop-up*, tornando possível que um site malicioso controle o conteúdo de uma janela *pop-up* de um outro site. O projeto *Common Vulnerabilities and Exposures* (cve.mitre.org) deu a essa falha o código [CAN-2004-1156](#).

Há um *bug* na maneira como o Firefox permite que os *plug-ins* adicionem conteúdo a um *frame*. É possível que uma página maliciosa induza o usuário a clicar em certos lugares para modificar configurações ou executar código arbitrário. O projeto *Common Vulnerabilities and Exposures* (cve.mitre.org) deu a essa falha os códigos [CAN-2005-0232](#) e [CAN-2005-0527](#).

Outra falha também grave: um atacante poderia explorar a forma como o Firefox mostra os domínios internacionais – como, por exemplo, [.co.uk](#), [.org.br](#) e [.edu.af](#). É possível que o agressor mostre uma URL válida, mas que aponte para outra página que não a correta. O usuário pensa que está vendo a página legítima mas está, na realidade, visitando uma impostora. O projeto *Common Vulnerabilities and Exposures* (cve.mitre.org) deu a essa falha o código [CAN-2005-0233](#).

Uma falha do tipo “espírito de porco”: a maneira como o Firefox trata os arquivos temporários dos *plug-ins* pode permitir que um usuário apague arquivos de outro. Por exemplo, um agressor local poderia criar links simbólicos na pasta `/tmp` que apontem para o diretório pessoal (`/home`) da vítima. Quando o Firefox for finalizado, os arquivos no `/home` da vítima são apagados. O projeto *Common Vulnerabilities and Exposures* (cve.mitre.org) deu a essa falha o código [CAN-2005-0578](#).

Os conversores de UTF-8 do Firefox também estão infestados de baratas. Um agressor poderia enviar uma cadeia de caracteres UTF-8 especialmente formulada ao conversor, levando à execução de código arbitrário. O projeto *Common Vulnerabilities and Exposures* (cve.mitre.org) deu a essa falha o código [CAN-2005-0592](#).

Nem o gerenciador de segurança de javascript do Firefox escapa. Se um usuário arrasta e solta um link malicioso para uma aba, o gerenciador é contornado. Com isso, é possível executar código remotamente ou conseguir informações sigilosas. O projeto *Common Vulnerabilities and Exposures* (cve.mitre.org) deu a essa falha o código [CAN-2005-0231](#).

O prompt de autenticação (usuário e senha) do Firefox também tem problemas. Quando o usuário entra em uma página privada e é instado a informar seu nome e sua senha, a caixa de diálogo é mostrada sobre a aba ativa, não importando qual aba disparou o pedido de nome e senha. Isso pode ser usado para enganar o usuário, fazendo com que ele dê o nome e senha de um site real, embora a caixa de diálogo tenha sido disparada por um site malicioso. O projeto *Common Vulnerabilities and Exposures* (cve.mitre.org) deu a essa falha o código [CAN-2005-0584](#).

Outro problema em caixas de diálogo: a janela *Salvar arquivo* permite que um site malicioso manipule o cabeçalho *Content-Disposition*, fazendo com que o usuário pense que está baixando um arquivo de tipo diferente do que ele realmente está baixando – por exemplo, o usuário pode ser levado a pensar que está baixando um MP3 quando na verdade é um executável do Windows. O projeto *Common Vulnerabilities and Exposures* (cve.mitre.org) deu a essa falha o código [CAN-2005-0586](#).

Também há falhas no sistema de auto-completar do Firefox, usando as setas de direção do teclado. Quando uma opção do auto-completar é selecionada, a informação é copiada no controle de entrada. Um site malicioso poderia roubar informações do usuário se puder induzi-lo a escolher certas opções. O projeto *Common Vulnerabilities and Exposures* (cve.mitre.org) deu a essa falha o código [CAN-2005-0589](#).

Não uma, mas várias falhas foram encontradas na rotina que mostra o ícone de sites seguros. Um site malicioso poderia

forçar a exibição do ícone seguro mesmo que os certificados de criptografia estejam incorretos. O projeto *Common Vulnerabilities and Exposures* (cve.mitre.org) deu a essa falha o código [CAN-2005-0593](#).

Há um *bug* na maneira como o Firefox mostra a caixa de diálogo de download. Um site malicioso poderia encobrir a URL a partir da qual o conteúdo está sendo baixado, fazendo o usuário acreditar que ele vem de um site confiável e conhecido – quando, na verdade, está vindo de um site suspeito. O projeto *Common Vulnerabilities and Exposures* (cve.mitre.org) deu a essa falha o código [CAN-2005-0585](#).

Mais um: há uma falha de interpretação no processamento das diretivas *xsl:include* e *xsl:import*. É possível que sites maliciosos importem folhas de estilo XSLT de um domínio atrás de um firewall, permitindo que um atacante obtenha informações sigilosas. O projeto *Common Vulnerabilities and Exposures* (cve.mitre.org) deu a essa falha o código [CAN-2005-0588](#).

Outra falha foi encontrada na maneira como o Firefox mostra a caixa de diálogo de confirmação de instalação. Um agressor poderia adicionar uma cadeia de caracteres no formato *usuario:senha*, escondendo o nome real do site de onde se está baixando o conteúdo a ser instalado. Com isso, o usuário pode ser levado a pensar que está instalando algo de um site confiável. O projeto *Common Vulnerabilities and Exposures* (cve.mitre.org) deu a essa falha o código [CAN-2005-0590](#).

Uma última falha foi encontrada na maneira como o Firefox mostra caixas de diálogo de downloads e de segurança. Um agressor poderia cobrir parte delas, podendo assim enganar o usuário e levá-lo a clicar em *Allow* (*Permitir*) ou *Open* (*Abrir*), o que poderia permitir a execução de código. O projeto *Common Vulnerabilities and Exposures* (cve.mitre.org) deu o código [CAN-2005-0591](#) a esta falha. ■

Referência no Gentoo: [GLSA 200503-10 / Firefox](#)

Referência no Red Hat: [RHSA-2005:176-11](#)

O Linux no mercado corporativo nacional

O Pingüim ganha espaço

Pesquisa da Fundação Getúlio Vargas mostra crescimento do Linux entre as empresas, mas não tanto quanto – ou na área em que – se esperava.

POR ALEXANDRE BARBOSA

Estamos acostumados a ler e acompanhar o crescimento do Linux, percebido pelo interesse da área de tecnologia sobre o tema, seja entre profissionais que se aperfeiçoam no uso da ferramenta, em empresas que provêem novas soluções e na adoção cada vez maior do Software Livre em seus vários desdobramentos nas organizações governamentais e mesmo em computadores pessoais ou em dispositivos com o Linux embarcado.

É natural perceber números cada vez maiores quando se trata da adoção do Linux em pesquisas de mercado. Então, é com um misto de confirmação de expectativas e banho de água fria que nos deparamos com a 16ª edição da *Pesquisa de Administração de Recursos de Informática da Fundação Getúlio Vargas*. Realizada pela Escola de Administração de Empresas de São Paulo, a pesquisa se baseou nas respostas de 1600 empresas brasileiras de médio e grande porte, 60% das quais pertencem às 500 maiores companhias em operação no País.

A pesquisa, composta de um formulário simples e diversificado, que cobre de sistemas de gestão empresarial a edição gráfica em uso nas empresas, envolve também os aplicativos de automação de escritório e sistemas operacionais presentes em servidores e nas estações de trabalho. “A pesquisa é propositalmente enviesada para retratar a realidade das médias e grandes empresas”, afirma o coordenador do levantamento, o professor Fernando Meirelles.

Os resultados apontam que as empresas estão gastando um percentual cada vez maior de seu faturamento em tecnologia da informação, tendo aplicado 5,1% no ano passado, e que existem pelo menos 24 milhões de computadores em uso no Brasil (somando PCs domésticos e corporativos). E o que ela diz sobre o Linux?

O Linux cresce, sim, mas não tanto e não da forma que se esperava. De acordo com o levantamento, o Linux responde hoje por 15% dos servidores principais das empresas, número este que está crescendo, mas abarcando um espaço que antes pertencia a outros sistemas Unix, em vez de outros sistemas proprietários. “A fatia Unix como um todo, entretanto, não está crescendo”, diz Meirelles.

Juntando esses sistemas Unix e Linux, a fatia é de 30% dos servidores contra 63% de equipamentos com algum “sabor” de Windows®, 4% com Novell e 3% dispersos entre outras plataformas operacionais. Na área de desktops, a coisa é pior: 98% dos PCs usa alguma variante de Windows® (a pesquisa não aponta se são versões legais ou não) contra 1% de sistemas baseados em Unix (Linux incluso).

A boa notícia? O Linux foi a plataforma com a maior taxa de crescimento no último ano, com 25% na comparação com 2003. A surpresa? A fatia da Microsoft também está crescendo: o S.O. da MS estava em 62% dos servidores em 2003.

Nos programas de automação de escritório, o Microsoft Office responde por 93% da base instalada nas empresas,

com alternativas como o OpenOffice.org e StarOffice respondendo por 6%. A novidade é que essa participação tem tendência ao crescimento, devendo chegar a 7% neste ano.

De acordo com o Meirelles, o Linux está crescendo, mas não vai dobrar de tamanho em poucos anos como alardeado por outras empresas de pesquisa. ■

Opinião» Pode ser que os defensores do Linux não gostem do que aponta o levantamento, com números bem conservadores em contraste com as hipóteses mais otimistas. Também pode ser verdade que o Linux esteja fazendo a revolução em empresas de pequeno a médio portes não envolvidas na pesquisa, o que poderia justificar o desvio estatístico. Independente dos questionamentos, a verdade é que a pesquisa da FGV lida com a maior amostragem dentre todos os levantamentos feitos no Brasil, o que mostra mais uma vez a necessidade premente, para os defensores do Linux, de apoiar a realização de estudos que mostrem o real crescimento do Software Livre no país. As empresas concorrentes não vão assistir passivamente ao crescimento do Linux; certamente executarão ações com o intuito de aumentar sua participação nos sistemas corporativos e governamentais.

E porquê o Linux não cresce nas médias e grandes empresas? Simples: estas já têm legado no software proprietário, o que, combinado a uma certa inércia quanto ao uso de aplicativos livres para automação de escritório e ao uso do Linux como sistema operacional para desktops, torna a manutenção da solução proprietária mais atraente, devido à continuidade dos investimentos. Outra razão é a falta de grandes casos de uso amplo, que poderiam dar mais confiança às empresas na hora de apostar no Linux em servidores e estações de trabalho. Os prestadores de serviço e desenvolvedores ainda têm muito trabalho a fazer na educação do mercado quanto às vantagens do uso de Software Livre em sua infra-estrutura tecnológica.

Fernando Meirelles - FGV



Notícias do Kernel

❑ Série 2.4 terá uma área para “Hot Fixes”

Após consultar Marcelo Tosatti, Willy Tarreau criou uma árvore chamada “Hot Fix”, que conterá correções agendadas para a próxima versão oficial do kernel 2.4. A razão para isso é que as versões oficiais podem demorar dois meses ou mais para ser lançadas. A árvore *-hf* seria um paliativo para quem precisasse das correções mais cedo. Entre as correções, teremos atualizações de segurança, correções para falhas críticas de corrupção de dados e *kernel panic*, remendos para conter *oops* e vazamentos de memória, soluções para funcionamento errôneo em geral, problemas com o sistema de compilação e com a documentação.

Essa é uma daquelas mudanças destinadas a uma minoria muito pequena de usuários, já que compromete o objetivo de manter a árvore 2.4 o mais estável possível. Toda a empolgação do pessoal da árvore 2.6 parece não contagiar Marcelo e não o desviam do mantra “2.4 estável, 2.4 estável, 2.4 estável” – mesmo que tenha sido persuadido a aceitar mais características portadas da conturbada versão 2.6 do que gostaria. Sua resistência a qualquer influência desestabilizante tem sido bem forte, no geral, e isso é comprovado pela criação de uma árvore separada como a *-hf* para conter a sanha modificatória na árvore principal. ■

❑ FUSE põe um pé dentro do kernel

Embora ainda não esteja incluído na árvore principal do kernel 2.6, o FUSE (*Filesystem in User Space* ou sistema de arquivos no espaço do usuário) foi aceito na árvore de Andrew Morton, a famosa *-mm*. Com isso, o número de usuários do FUSE aumenta bastante e, com ele, a re-

alimentação sobre problemas, falhas e desempenho. Grande vitória para os desenvolvedores do FUSE, já que o caminho para a inclusão no kernel 2.6 pode ser comparado a uma batalha campal contra um inimigo fortificado colina acima. Seu primeiro obstáculo foi Linus Torvalds em pessoa, que considera existirem problemas inerentes à idéia de se colocar o controle dos sistemas de arquivos nas mãos do usuário. Linux põe esses problemas no mesmo balaio dos que infestam os microkernels (lembra das “brigas” com Andrew Tannenbaum?) como as intermináveis camadas de abstração e a compartimentalização das funcionalidades. Mas ele ainda pode ser convencido e a presença do FUSE na árvore *-mm* pode encurtar o caminho e responder a muitas de suas objeções. ■

❑ O pesadelo da documentação

Pelo menos dois documentos significativos foram trazidos à luz do dia recentemente. O primeiro, de Josh Aas, documenta o agendador (*scheduler*) da CPU para o kernel 2.6.8.1 – aparentemente um kernel ainda bastante popular em sistemas de produção. O empregador de Josh, a empresa SGI, detém os direitos autorais sobre o documento, lançado sob a licença GNU FDL – GNU Free Document License.

Noutra frente, Stelian Pop liberou um *Como Fazer (HOWTO)* descrevendo o rastreamento e controle de versões do kernel usando o sistema de controle de versões *Subversion*. Esse documento (ou talvez o debate sobre ele na lista de discussão do Kernel) trouxe à baila opiniões iradas e apaixonadas, já que colocou em cheque – mais uma vez – se é apropriado ou não usar o BitKeeper para um projeto livre como o kernel do Linux.

Qualquer desenvolvedor que queira contornar o BitKeeper usando o *Subversion*, o *arch* ou qualquer outra ferramenta tende a exigir que a empresa BitMover ofereça um suporte cada vez melhor à sua alternativa do coração. Mesmo que Larry McVoy, CEO da BitMover, tenha a intenção de ir até o limite do possível para auxiliar os usuários do BitKeeper, ele tem sido um verdadeiro “paredão” intransponível quando se trata de ajudar os competidores do BitKeeper. Sua postura não pode ser censurada, afinal ele construiu um negócio sólido ao redor do BitKeeper e o mundo do software livre ainda não foi capaz de criar nada tão bom que possa substituí-lo – mesmo depois de anos de luta. É fato que projetos como o *arch* tem um grande futuro, mas há um abismo gargantual entre as melhores alternativas livres e o BitKeeper. Mesmo o *arch* mostrou recentemente que, devido a falhas de projeto, está muito aquém do que um empreendimento tão gigantesco quanto o kernel Linux precisa. ■

❑ Arquivos além da imaginação

O kernel é mesmo uma criatura bizarra. Quando funciona, é a mais bela das feras. Quando entra em parafuso, pode se tornar o monstro mais horrendo do espaço sideral. Entretanto, algumas pequenas falhas são tão estranhas que nada podemos fazer a não ser gargalhar.

Wichert Akkerman experimentou uma dessas falhas recentemente. Com um sistema de arquivos *ext3* rodando sob um kernel 2.6.10-ac10 (de Alan Cox, portanto), ele notou que o comando *df* informou que a taxa de ocupação do disco estava em 101% e que o número de blocos disponíveis era de -73786976294838127736. Não parece ter havido nenhuma perda de da-

dos ou corrupção e nada mais apresentou problemas. A informação maluca foi-se embora depois de um `fsck` redentor.

Sem informações suficientes para tentar um diagnóstico, essa falha deve ser catalogada para referência futura caso algo similar seja informado. Se o problema não se deveu a alguma falha de hardware, Wichert deve ter disparado um *bug* de alguma forma. Como ele o fez, entretanto, ainda é um mistério.

Um problema menos misterioso mas não menos embaçante foi descoberto por Mitch Williams. Ele descobriu que

se abríamos um arquivo do SysFS para edição e modificarmos algo, a modificação não será inserida no ponto em que a escrevemos. Em vez disso, o arquivo será substituído por outro de mesmo nome contendo **apenas** o novo dado inserido. Da mesma forma, a tentativa de concatenar dois arquivos no SysFS resulta, em vez da combinação dos dois arquivos, na substituição do primeiro pelo segundo. Isso foi um choque para caras como Greg Kroah-Hartman, que colocou meses de trabalho e noites sem dormir no SysFS.

Aparentemente, a maneira como o SysFS é usado não é prejudicada pela falha citada. As pessoas não costumam modificar os arquivos do SysFS, mas substituem completamente seu conteúdo. Ainda assim, esse comportamento errático deve ser consentado: o SysFS precisa, obrigatoriamente, funcionar como qualquer sistema de arquivos comum. Mitch enviou uma correção para consertar esse comportamento macabúzio e Greg se ofereceu para auxiliar Mitch a tornar o patch aceitável para inclusão na árvore principal. ■

❑ Novos planos para a numeração das versões

Reviravolta na novela! O processo de desenvolvimento do kernel está trocando a pele novamente, revelando uma nova encarnação de sistemas de numeração. Há dois aspectos principais e nenhum deles foi registrado em cartório ainda (aliás, sequer foram escritos a lápis em papel de padaria...). O primeiro é que Linus Torvalds e Andrew Morton estão considerando o retorno do modelo versão par/versão ímpar para diferenciar novamente a versão de desenvolvimento da versão estável. Entretanto, haveria um novo elemento no enredo: em vez de apenas o chamado *número menor* ter seu significado ditado pela diferenciação par/ímpar, sugeriu-se que todos os três números sigam essa lógica – o *número maior*, que representa a versão principal do kernel, o *número menor*, associado às versões intermediárias ou *edições*, e o chamado *número de patch*, que indica as revisões e correções feitas numa determinada série (a *série* do kernel é indicada pelos números maior e menor, por exemplo, série 2.6).

Da forma como foi proposta por Linus, o significado de um dígito ímpar aumenta de acordo com sua posição – quanto mais à esquerda, mais importante ele é. Portanto, se todos os números forem pares (2.6.14 por exemplo), isso representaria uma semana e pouco de esforço para estabilização do kernel. Se o número de patch for ímpar (2.6.13), o objetivo principal é a estabilidade, mas esse kernel é o resultado de grandes mudanças (ou seja, novas funcionalidades foram adicionadas e modificadas) e será o produto de um mês ou dois de trabalho. Se o número de edição (o número menor) for ímpar (2.7.10), esse kernel é objeto de mudanças tão profundas que podem desestabilizar o código por muitas edições no período de um ano ou dois. E se o número principal for ímpar (3.0.0) houve quebra geral de compatibilidade e o kernel está sendo praticamente reescrito do zero. Também significa que Linus está no hospital recuperando-se de estafa ou de algum distúrbio psiquiátrico.

Não nos parece que a bênção final tenha sido dada, mas é muito raro que esse tipo de mudança tenha um ponto de partida tão claro como o que Linus e Morton ofereceram desta vez. Linus prefere uma abordagem mais iterativa, difícil para os jornalistas entenderem, mas que facilita a vida dos desenvolvedores do kernel.

O segundo aspecto novo do desenvolvimento do kernel é algo que vinha cozinhando em fogo lento já há algum tempo: a criação de um esquema de numeração de **quatro** números, *X.Y.Z.A* – chamada por Linus de “*the sucker tree*” (a árvore dos otários) pois só um otário (*sucker*) concordaria em manter uma barbaridade dessas. A razão é óbvia: depois de todos os senões e restrições discutidos na LKML (a lista de discussão do kernel) essa árvore vai ser um pesadelo de se manter. Afinal, o “salário” é baixo, as honorarias são quase inexistentes e haverá uma multidão de usuários irados vociferando contra o mantenedor e pedindo sua cabeça.

Linus quer um controle rígido sobre a árvore *X.Y.Z.A* para assegurar que apenas as alterações adoradas como deusas por todos os desenvolvedores sejam incluídas. Os patches devem ser *apenas* correções de erros com nível de pureza próximo dos 100%, o que quer dizer que os desenvolvedores “espertos” que costumam incluir novas funcionalidades (não testadas) nos patches serão escorraçados da árvore proposta. Haverá um período de dois dias para apreciação do patch pelos demais desenvolvedores. Apenas se sobreviver a esse período o patch pode ser aceito. Linus irá puxar todos os patches da árvore dos otários para a sua própria, qualquer que seja o estágio de desenvolvimento ou instabilidade que a árvore de Linus esteja experimentando. Os usuários terão muito mais confiança na estabilidade do kernel se usarem a versão da “árvore dos otários”, deixando a árvore principal para os malucos dos desenvolvedores enfiarem seus códigos cheios de doideiras barulhentas. Greg Kroah-Hartman tornou-se um dos mantenedores principais da “árvore dos otários” e liberou a versão 2.6.11.1 com três pequenas correções para falhas no 2.6.11. ■



Haroldo Hoffmann - In Press Brodeur

Entrevista com Haroldo Hoffmann e César Taurion

Pingüim de casaca azul



César Taurion - In Press Brodeur

Referência mundial em vários segmentos da tecnologia da informação, a IBM também é considerada um dos vetores quando o assunto é Linux. A empresa é uma das organizações privadas que mais investe no desenvolvimento de produtos em plataforma aberta, já tendo investido mais de US\$ 4 bilhões em pesquisa e desenvolvimento nos últimos dois anos, e também uma das que mais lucra com Open Source: só a divisão de servidores da série iSeries deve atingir US\$ 2 bilhões em faturamento com Linux neste ano, contra US\$ 1,1 bilhão alcançados no ano passado em todo o mundo, boa parte vindo de empresas de pequeno e médio porte. Para entender melhor os movimentos da companhia no Brasil, a Linux Magazine conversou com Haroldo Hoffmann, coordenador das ações de Linux da IBM no país, e Cesar Taurion, gerente de novas tecnologias.

POR ALEXANDRE BARBOSA

Linux Magazine» No que consiste a estratégia de Linux da IBM para o Brasil?

Haroldo Hoffmann» Temos muitas ações em andamento, divididas entre as diversas iniciativas desenvolvidas pela IBM nos mais variados mercados. São ações que compreendem o apoio ao desenvolvimento de novas soluções, ferramentas de integração e outros projetos. Também contamos com o LIC, o *Linux Integration Center* (Centro de Integração Linux), um espaço criado pela IBM Brasil em 2004 onde nossos clientes podem desenvolver provas de conceito, ou seja, podem simular as condições existentes em empresas reais comprovando a viabilidade da implementação do Linux e seu desempenho sem correr o risco de comprometer suas operações. Esse espaço foi ampliado recentemente e agora não só os clientes podem rodar suas aplicações: temos também a presença de parceiros que podem mostrar soluções de CRM, financeiras ou outras aplicações de missão crítica rodando em Linux.

Cezar Taurion» Existem diversas soluções em operação nesse espaço, confirmando a competência do Linux em diversos campos, com ferramentas de Voz sobre IP (VoIP), RFID (*Radio Frequency Identification*), soluções para caixas eletrônicas (ATMs) e diversas outras tecnologias que estão à disposição dos interessados e que podem ver, ao vivo, que os possíveis problemas de integração podem ser resolvidos, além de conhecer a aplicabilidade dessas novas tecnologias.

LM» De que forma a estratégia “on demand” (de computação sob demanda, oferecida como um serviço) da IBM se relaciona com o Linux?

HH» A IBM acredita em padrões abertos e o apoio ao Linux é uma parte fundamental desse pilar, construído para proporcionar valor e prestar melhores serviços ao cliente. Acreditamos no potencial da plataforma, que é independente do hardware e atende às nossas perspectivas de negócio. Nesse sentido temos alguns serviços de *outsourcing* em que o Linux desempenha um papel vital. Mas para que os clientes tenham confiança e repassem seus processos é preciso que conheçam nossa infra-estrutura. Para tanto, é vital contarmos com o Centro de Competência em Linux, uma instalação onde os clientes podem fazer testes de compatibilidade de sistemas em máquinas rodando Linux, tudo isso obedecendo a acordos de nível de serviço – os chamados SLAs (*Service Level Agreements*) – além de poderem portar soluções que rodam em outras plataformas para que operem sobre o Linux.

CT» A migração para o universo *on demand* se relaciona com necessidades surgidas de um ambiente de maior competição na sociedade pós-industrial. Há setores de atividade em que há mais oferta do que demanda por produtos, por exemplo. Junte a isso o fato de que as cadeias de valor são complexas e pedem níveis elevados de integração para mitigar custos e ganhar agilidade. A chave para que isso aconteça é

o fluxo de informação. É aí que se encaixa a estratégia *on demand*, um conjunto de novas idéias com padrões abertos, aplicações encapsuladas, escaláveis e que podem ser contratadas conforme a necessidade, trazendo mais rapidez para as operações corporativas, atendendo às necessidades mais dinâmicas das empresas. Isso sem contar que o *on demand* possibilita o compartilhamento de recursos e custos entre parceiros de negócio.

LM» *Que outras tecnologias estão sendo amparadas pelo Linux?*

CT» Duas tendências futuras são o *grid computing*, com a computação distribuída e o uso de recursos latentes nas organizações. Nas empresas, há muitas máquinas que são subutilizadas. A “computação de grade” permite identificar esses recursos e alocá-los para aplicações que vão otimizar a base instalada de ativos de cada companhia. É aplicar a analogia da Internet para o universo da computação distribuída. E já existem desdobramentos do grid, como o *grid interno*, em que há um sistema distribuído dentro de uma rede ou então um *extra grid*, que, a exemplo das *extranets*, aplica o conceito de grade entre parceiros. Outra inovação são os sistemas de *autonomic computing*, ou computação autônoma, sistemas que têm a capacidade de se autogerenciar, de reparar-se sem a necessidade de intervenção humana, o que é muito interessante para empresas que precisam dispor de ferramentas tecnológicas vitais para o negócio, sejam elas sistemas de processamento financeiro ou aplicações de VoIP. São essas as visões que a IBM tem hoje do maior aproveitamento da tecnologia e estamos seguros de ser a única companhia capaz, hoje, de oferecer essa visão com o suporte tecnológico necessário. Falando em Linux, claro, o espectro de possibilidades é muito maior: existem iniciativas para promover seu uso em produtos eletrônicos, como é a iniciativa CELinux (CE, de *Consumer Electronic*), que visa embutir Software

Livre nos mais variados equipamentos. É esse um dos grandes valores do Linux: sua versatilidade, que não é encontrada em nenhuma outra plataforma.

LM» *A IBM tem um desafio de imagem ao propor novas idéias para o público em geral. Como a empresa lida com esse desafio?*

HH» Garantimos nossa participação em todos os eventos significativos envolvendo o Linux, do CONIP ao Fórum Internacional de Software Livre, passando por eventos internacionais. Temos uma campanha mundial de divulgação do Linux; cultivamos um relacionamento próximo com as Universidades e apoiamos as ações de parceiros locais que ajudam a fomentar o progresso da plataforma, além de apoiar o avanço do Linux nas empresas e no Governo.

LM» *Que desafios as empresas brasileiras que lidam com Linux têm de vencer para prover melhores serviços?*

CT» Ainda há quem imagina o Linux como coisa de criança, em cima da cultura do *geek*, que era válida no início. Não é o caso agora. As empresas têm uma visão mais madura do assunto, o que obriga os prestadores de serviço a se profissionalizar. Lidar com Linux é uma questão de negócio.

HH» Vemos que a entrada de grandes empresas no negócio do Linux está promovendo um aperfeiçoamento nos níveis de serviço como um todo. Também há o interesse cada vez maior das empresas que querem entender o mundo Linux, mesmo que não pensem em migrar em breve. Temos mundialmente mais de 600 profissionais trabalhando exclusivamente na qualidade do Linux e elaborando melhorias para o kernel. 80% delas têm sido incorporadas ao Linux. Temos parte dessa equipe no Brasil, com 12 pessoas na Unicamp, em São Paulo, com 10 bolsistas patrocinados pela IBM trabalhando sob a coordenação de dois funcionários da IBM. Lá também mantemos um mainframe em Linux servindo como uma central

para novos projetos, além de apoiarmos a Universidade de Brasília com o suporte ao Centro de Difusão de Tecnologia, que é um núcleo importante para as iniciativas de Linux no âmbito governamental.

CT» O mainframe é interessante, já que pode rodar várias instâncias do Linux simultaneamente, abrigando várias máquinas virtuais – o que é uma forma de abastecer as iniciativas de computação sob demanda que citei anteriormente.

LM» *Qual é a importância do relacionamento com os parceiros?*

HH» É fundamental, por isso investimos nas parcerias com empresas que atuam em vários mercados verticais. Rodar Linux no servidor já é lugar-comum. O desafio agora consiste em abraçar aplicações de ERP, CRM, soluções financeiras e outras ferramentas corporativas. É onde entram alguns parceiros como a Datasul, Microsiga, XSeed, entre outras, que mostram que há uma infinidade de aplicações de missão crítica rodando no Linux com maior disponibilidade e custos finais menores.

LM» *Hoffmann, como coordenador da área de Linux, qual seu principal desafio?*

HH» É quebrar um paradigma de venda de produtos existente na companhia. A IBM sempre foi vista como uma empresa que vendia hardware e agora está migrando para a oferta de serviços. Isso obriga os vendedores a se aproximar das atividades dos clientes, entender melhor suas necessidades e buscar formas de aplicar a tecnologia como uma alavanca para sua produtividade. Assim, minha missão é consolidar as diversas ofertas existentes em hardware, software e serviços e dar ao cliente uma forma de resolver seus problemas com mais agilidade. O desafio é fazer as várias equipes e envolvidas terem essa visão de conjunto, envolvendo o Linux e todo o ecossistema de soluções que giram em torno da plataforma e que podem beneficiar o cliente final. ■

Mundo livre em revista



☐ Sun indica membros do conselho administrativo do OpenSolaris

A Sun Microsystems, prossequindo com o processo de criação do projeto OpenSolaris, designou os membros do conselho administrativo da comunidade (CAB), ao qual cabe a função de coordenar o desenvolvimento da versão de código aberto do sistema operacional da empresa. Os escolhidos representarão tanto a empresa e a comunidade Open Source quanto a comunidade OpenSolaris.

Os escolhidos da Sun são Roy Fielding, um dos fundadores da *Apache Software Foundation*, Simon Phipps, o *Chief Technology Evangelist* da Sun e Casper Dik, *Senior Staff Engineer* da empresa. Da comunidade OpenSolaris “piloto” foram escolhidos Al Hopper, consultor de tecnologia da empresa *Logical Approach* e Rich Teer, consultor de sistemas Solaris e autor do livro *Solaris Systems Programming*.

Segundo Simon Phipps, em uma entrevista publicada pela Sun, “a composição equilibrada e ao mesmo tempo heterogênea do conselho, que deve ser independente e comprometido com integridade e inovação, é exatamente o que a comunidade OpenSolaris necessita para embasar suas decisões”. De acordo com Phipps, a comunidade de desenvolvedores em torno do OpenSolaris já existe e o projeto está capacitando milhares de novos participantes e colaboradores, além de incrementar o raio de ação tanto da metodologia quanto do movimento em torno de sistemas de código aberto.

Além da versão comum do Solaris 10, que para a empresa já representa uma história de sucesso, a Sun iniciou o projeto OpenSolaris, sob o qual o código fonte do sistema operacional da companhia deve ser liberado. A licença escolhida para isso foi a *Common Development and Distribution License* (CDDL), sobre a qual há diferentes opiniões entre os desenvolvedores e protagonistas do “universo” Open Source. A primeira porção de código já aberta do Solaris 10 é a que contém o *Software Dynamic Trace* (DTrace), conforme registramos em notícia à página 16 da sexta edição da *Linux Magazine*. O DTrace é um programa que analisa as características de execução de aplicativos e do ambiente em que eles rodam em tempo de execução, procurando por gargalos no desempenho do sistema ou da rede que possam ser minimizados. ■

<http://www.sun.com/>

<http://www.opensolaris.org/>

<http://www.sun.com/aboutsun/media/presskits/osbc/bios.html>

<http://roy.gbiv.com/>

<http://www.webmink.net/>

<http://www.sun.com/aboutsun/media/presskits/osbc/bios.html#dik>

<http://www.sun.com/aboutsun/media/presskits/osbc/bios.html#hopper>

<http://www.rite-group.com/rich/>

http://www.sun.com/aboutsun/media/features/opensolaris_qa.html

<http://www.opensolaris.org/license/index.html>

<http://www.sun.com/bigadmin/content/dtrace/>

☐ Linux em servidores virtuais Microsoft

Steve Ballmer, por ocasião da *Microsoft Management Summit* em Las Vegas, anunciou que a Microsoft irá oferecer suporte oficial para o Linux como sistema hospede do Virtual Server 2005.

O suporte ao Linux no Virtual Server deverá ser melhorado com um *Service Pack*, disponível atualmente em uma versão de testes na homepage do produto. O Linux já “roda” no software de virtualização da Microsoft. Entretanto, até agora a empresa não fornecia suporte oficial ao sistema.

O Virtual Server é a versão para servidores do software de emulação e virtualização VirtualPC, que permite a execução de diferentes sistemas operacionais em um único computador. A Microsoft adquiriu o software em 2003, com a compra da tecnologia da Connectix. No Virtual Server há recursos extras de administração de máquinas virtuais, tais como uma interface de administração web através da qual os sistemas virtuais podem ser iniciados e desligados. A empresa considera o produto uma peça adicional na sua estratégia *Dynamic Systems Initiative*, para a qual anunciou recentemente extensões administrativas para redes de grande porte. ■

<http://www.microsoft.com/windowserversystem/virtualserver/default.mspx>

<http://www.microsoft.com/windows/virtualpc/default.mspx>

<http://www.microsoft.com/management/default.mspx>



❑ Novell lança SUSE LINUX 9.3

A Novell lançou a versão 9.3 do SUSE LINUX. As entregas na Europa já começaram e no website da empresa já é possível encomendar a nova versão.

Como de costume, a oferta de software da distribuição foi atualizada: kernel Linux 2.6.11, KDE 3.4, Gnome 2.10, Firefox 1.0, um pré-release do OpenOffice.org 2.0, Evolution 2.0 e Gimp 2.2 são apenas alguns dos aplicativos disponíveis na distribuição que se encontram na versão mais recente. Novidades adicionais, de acordo com o fabricante, são o suporte ao iPod e melhor compatibilidade com notebooks, a presença do software de telefonia IP Linphone e a integração da solução de virtualização Xen. O SUSE LINUX 9.3 está disponível para processadores x86 de 32 e 64 bits da Intel e da AMD.

As imagens ISO da nova versão para download ainda deverão demorar um pouco – até o fechamento desta edição só era possível encontrar a versão anterior, a 9.2, no site da empresa. O mesmo vale para os servidores FTP da SUSE, nos quais, entretanto, já é possível encontrar atualizações para a nova versão, tais como o suporte a MP3 para os programas multimídia disponíveis na distribuição. ■

<http://www.novell.com/>

<http://www.suse.com/>

<http://xen.sf.net/>

<http://www.novell.com/products/linuxprofessional/downloads/ftp/eval.html>

<http://ftp.suse.com/pub/suse/i386/update/9.3/>

❑ Desenvolvimento do Linux a partir de agora sem BitKeeper

Linus Torvalds, criador do Linux, de acordo com mensagem postada na lista de discussão do kernel Linux, vai deixar de usar a ferramenta *BitKeeper* no desenvolvimento do kernel. A razão para o abandono da ferramenta foi o anúncio da descontinuação do cliente *BitKeeper* gratuito.

No passado, Linus foi freqüentemente criticado (para usar uma palavra suave) por usar um programa comercial para o desenvolvimento do kernel Linux, que é sabidamente Software Livre – ainda mais porque a versão do *BitKeeper* para projetos de código aberto sempre foi limitada, em comparação com a versão comercial. O próprio fundador do projeto GNU, Richard Stallman, foi um dos maiores críticos da atitude de Linus, ressaltando sempre que o *BitKeeper* era um programa gratuito, mas não era livre e, desse modo, não era passível de modificações, mesmo quando estas se fizessem necessárias.

Apesar das críticas, desde a introdução do programa para administração do desenvolvimento do kernel em 2002, o sistema se estabeleceu entre os desenvolvedores de subsistemas do Linux como ACPI, Firewire, SATA, IDE, PCI, rede, USB, DRM e Itanium. Linus está agora à procura de uma alternativa de código aberto para o sistema, que – à sua semelhança – trabalhe com um modelo descentralizado de administração de repositórios de códigos fonte. Entre os candidatos aventados por ele estão o projeto *monotone*, mas no âmbito da discussão já foram também

citados os projetos *arch* e *bazaar-ng*. O uso do CVS e do (moderno) *subversion* estão fora de questão para Linus – a própria equipe de desenvolvimento do *subversion* listou os motivos pelos quais o programa não é adequado para o desenvolvimento do kernel e sugeriu o uso do *svk*, ferramenta de gestão descentralizada de código fonte baseada no *subversion*.

Linus e outros desenvolvedores do kernel agradeceram à empresa que desenvolveu o *BitKeeper*, a *BitMover*, bem como ao seu diretor-presidente, Larry McVoy, pelos serviços prestados até agora, ressaltando que a ferramenta contribuiu para simplificar e acelerar o desenvolvimento do Linux em diversas áreas. Agora resta encontrar uma alternativa para gestão de código fonte que seja capaz de substituir o programa, que possa dar cabo da tarefa hercúlea de gerenciar a árvore de diretórios gigante do kernel. ■

<http://article.gmane.org/gmane.linux.kernel/293914/>

<http://www.bitkeeper.com/>

<http://www.stallman.org/>

<http://www.uwsg.iu.edu/hypermail/linux/kernel/0202.0/0989.html>

<http://www.venge.net/monotone/>

<http://thread.gmane.org/gmane.linux.kernel/293914/>

<http://www.gnu.org/software/gnu-arch/>

<http://bazaar-ng.org/>

<http://subversion.tigris.org/>

<http://svk.elixus.org/>

<http://www.bitmover.com/>

❑ Mandrake + Conectiva = Mandriva!

A Mandrakesoft, produtora da distribuição Linux Mandrakelinux, mudou de nome: devido à fusão com a empresa brasileira Conectiva, desenvolvedora da distribuição Linux de mesmo nome, a empresa passará a ser chamada de Mandriva. Conforme entrevista dada por Jacques Rosenzvaig, CEO da Conectiva, à página 14 da sétima edição da Linux Magazine, a nova nomenclatura “representa melhor o novo conceito de federação de empresas globais”, advindo da fusão das duas empresas.

Um outro motivo para a mudança de nome seria o longo conflito na justiça entre a Mandrakesoft e a a Hearst Cor-

poration, detentora da marca Mandrake, relativa ao mágico, um conhecido personagem dos quadrinhos. A mudança de nome, além de ser uma solução definitiva para o conflito, se estende também aos nomes dos produtos e à presença da empresa na Internet. Desse modo, a Mandrakelinux passa a ser Mandriva Linux, e a marca Conectiva Linux deixa de existir. A empresa também publicou novos logotipos. ■

<http://www.mandriva.com/>

http://www.kingfeatures.com/pressrm/rel_157_24_2_2004.htm

<http://www.hearstcorp.com/>

<http://www.mandriva.com/company/press>



❑ Lançada versão 5.04 do Ubuntu-Linux

Os desenvolvedores do Ubuntu-Linux lançaram a segunda versão de sua distribuição Linux projetada para uso no desktop, que, seguindo o esquema de numeração de versões adotado (último dígito do ano e mês de lançamento da versão, separados por um ponto), recebeu a numeração 5.04. A distribuição, baseada no Debian GNU/Linux e no ambiente desktop GNOME, recebeu o codinome *Hoary Hedgehog* e está disponível para download em diversos servidores espelho. Além da versão de 32 bits para PCs comuns, há também variantes para PowerPC e para sistemas baseados nas arquiteturas de 64 bits da AMD (AMD64) e da Intel (EM64T). Adicionalmente, há também um *live CD* com a distribuição para download que, à semelhança do Kurumin ou do Knoppix, pode ser usado para um teste “sem compromisso”, ou seja, sem necessidade de instalação. A nova versão do Ubuntu baseada no KDE, chamada Kubuntu, também chegou à versão 5.04.

O Ubuntu 5.04 vem com praticamente o que há de mais novo em software livre: kernel Linux 2.6.10, servidor gráfico X.org 6.8.2, ambiente desktop Gnome 2.10 e navegador Mozilla Firefox 1.0.2. Uma nova versão da distribuição é lançada a cada 6 meses; atualizações de segurança ficam disponíveis por 18 meses após o lançamento de cada versão. O número de aplicativos que vêm no CD de instalação é muito menor do que a maioria das outras distribuições oferece, refletindo a mentalidade do projeto de fa-

cilitar as escolhas para o usuário, incluindo apenas os melhores aplicativos em cada categoria. Entretanto, um número enorme de aplicativos pode ser facilmente instalado via rede – desde que se disponha de banda larga, naturalmente.

Nos bastidores do Ubuntu (uma palavra que, entre outras coisas, significa “humanidade para com os outros”) está a empresa Canonical, de propriedade do multimilionário sul-africano Mark Shuttleworth (que, depois do empresário californiano Dennis Tito, foi o segundo “turista espacial”). A distribuição é disponibilizada gratuitamente – a Canonical deseja faturar com os serviços de suporte ao Ubuntu.

A primeira versão do Ubuntu, sobre a qual falamos na sétima edição da Linux Magazine (ver artigo à página 35), foi lançada em outubro de 2004, mas mesmo antes do lançamento a distribuição começou a “causar sensação”, quando a Canonical passou a empregar uma série de personalidades do universo Open Source. O lançamento da primeira versão trouxe à distribuição uma legião de fãs. Na longa lista de mais de 400 distribuições Linux disponível em lwn.net/Distributions/, o Ubuntu Linux é citado junto com as distribuições já estabelecidas no mercado, como Debian GNU/Linux, Fedora, Red Hat e SUSE LINUX, entre outras. ■

<http://www.ubuntulinux.org/>

<http://www.ubuntulinux.org/download/>

<http://kubuntu.org/>

<http://www.canonical.com/>

❑ Agora é pra valer: Adobe libera a versão 7 do seu leitor de PDFs para Linux

Primeiro ela foi lançada, depois não era bem isso, mas agora é pra valer: a Adobe liberou oficialmente a sétima versão para Linux do seu leitor de documentos no formato PDF: o Adobe Reader 7. A confusão ocorreu porque a empresa lançou um *pre-release* da nova versão para clientes na Holanda.

Com o lançamento da versão 7 do Adobe Reader, a empresa está disponibilizando novamente uma versão atual do programa para Linux, já que a última versão disponível para o sistema de código aberto foi a 5. A nova versão do leitor de PDFs gratuito, da mesma forma que as versões anteriores, permite, além da leitura e impressão de PDFs, o preenchimento de formulários eletrônicos. Modificações nos arquivos PDF são possíveis, desde que o autor original do documento tenha dado permissão para tanto. A nova versão pode exibir todos os documentos que utilizem as funções do Adobe Acrobat 7 – mesmo as avançadas.

Atualmente, somente a versão em inglês do Reader 7 está disponível para download.

<http://www.adobe.com/aboutadobe/pressroom/pressreleases/200504/041205LINUX.html>

<http://www.adobe.com/products/acrobat/readstep2.html>



❑ Linux realmente pequeno

A fabricante alemã Kleinhenz Elektronik anunciou que irá iniciar as vendas de um dispositivo movido a Linux pouco maior que um conector RJ-45, usado em portas Ethernet. Na verdade, ele até se parece com um conector RJ-45. Com um tamanho de 19x19x36 mm, o novo dispositivo faz outros sistemas com Linux embarcado parecerem mainframes. O minúsculo aparelho, que vai ser comercializado com o nome “Picotux”, é baseado no hardware DigiConnect ME, da Digi International.



De acordo com a Kleinhenz, o Picotux usa o kernel 2.4.27, portado para a plataforma Digi-connect ME. O sistema Picotux inclui uma

porta Ethernet e está pronto para ser ligado a uma rede. A empresa diz que este pequeno sistema Linux será comercializado como uma ferramenta para prototipação e produção de aplicativos embarcados, mas vamos esperar para ver que outros usos a comunidade de usuários, sempre criativa, descobre para o Picotux.

<http://www.picotux.com>

❑ Netline lança a versão 5 do servidor Open-Xchange

A Netline Internet Service liberou a versão comercial 5 do seu servidor de colaboração Open-Xchange. É a primeira solução de colaboração de código aberto com uma proposta de serviço e manutenção. O sistema é expansível utilizando módulos e compatível com todos os clientes *groupware* comuns. De acordo com a empresa, a idéia é permitir a migração para a arquitetura de código aberto, sem que o usuário final precise mudar seu cliente de email ou de colaboração (Outlook, Evolution ou Kontact).

O *Open-Xchange Server 5* contém todos os recursos da versão 0.8 do projeto de código aberto de mesmo nome e que foi liberado pela Netline sob a GPL em agosto de 2004. Além dos recursos da versão livre, a versão comercial contém, entre outras coisas, “conectores” para o Outlook e o Palm, uma interface web de administração e manuais para administradores e usuários, bem como cinco anos de garantia de disponibilidade de atualizações e serviços, além de suporte para a instalação do SUSE LINUX Enterprise Server 9, para o qual o sistema está disponível imediatamente. O suporte para as versões 3 e 4 do Red Hat Enterprise Server deve estar disponível a partir do final de maio. Há duas versões do sistema: a versão *Small Business*, de 5 a 25 usuários, e o *Advanced Server*, para mais de 25 usuários. A empresa não informa preços para o mercado brasileiro, mas coloca um formulário à disposição para compra no qual indica que os preços serão informados por email.

<http://mirror.open-xchange.org/ox/EN/product/>

❑ HP adota Ubuntu Linux como distribuição para notebooks

O uso de Linux em notebooks de hardware atual está sempre sujeito a algumas limitações, devido a questões de falta de suporte deste ou daquele recurso de hardware. A Hewlett-Packard decidiu dar um jeito nisso: uma versão especial do Ubuntu Linux deverá em breve oferecer suporte completo ao hardware de diversos modelos de notebook da empresa.

De acordo com a HP, para os modelos nx6110, nc6120, nc6220, nc6230 e nc6000, além do subnotebook nc4200 – oficialmente não suportado, mas equipado com o mesmo hardware dos anteriores – LAN, WiFi, modem e placa de som, PCMCIA, IR, FireWire e Bluetooth (desde que pré-instalados de fábrica) deverão funcionar totalmente no Linux. O suporte à aceleração 3D via hardware para a placa de vídeo Intel Graphics Media Accelerator 900 (a mais comum nos notebooks), integrada ao novo chipset Centrino Mobile 915, deverá ser disponibilizado pela distribuição. O recurso de operação *dual head* já está inclusive funcionando. A HP se preocupa em especial com o suporte completo aos recursos ACPI – da leitura do estado da bateria, passando pelo controle da frequência do processador, até os recursos *suspend-to-disk* e *suspend-to-RAM*, este último ainda em desenvolvimento para o Linux. Teclas de atalho especial (*hot keys*), como as de controle de nível de volume, e o “berço” (*docking station*) dos notebooks poderão ser utilizados com o Linux.

A HP não entregará o sistema pré-instalado, mas também não forçará o cliente a “engolir” uma licença do Windows® desnecessária: as máquinas virão pré-instaladas com o FreeDOS e um CD com a versão customizada do Ubuntu fará parte do pacote.

O Ubuntu é uma distribuição Linux baseada no famoso Debian GNU/Linux, mas é caracterizado por uma instalação simples com reconhecimento automático de hardware melhorado, além de otimizações para o uso no desktop e uma melhor usabilidade, advinda de uma boa pré-configuração do sistema. À exceção de drivers especiais e algumas customizações, a distribuição Linux resultante da colaboração com a HP deverá permanecer o Ubuntu “de sempre”, de modo que os mecanismos de atualização e instalação de aplicativos e de aplicação de correções de segurança (ambos usando o APT) poderão continuar a ser usados. Uma análise detalhada do Ubuntu pode ser encontrada à página 35 da sétima edição da Linux Magazine, que trouxe também a distribuição sul-africana no CD do mês. ■

<http://www.hp.com.br/>

<http://www.ubuntu.com/>

<http://www.freedos.org/>

<http://www.debian.org/>

❑ Liberada versão 4 da GNU Compiler Collection (GCC)

Mark Mitchell liberou a quarta versão da *GNU Compiler Collection* (GCC). A versão 4 traz como novidade principalmente um grande número de opções de otimização. Um exemplo disso é a introdução do *framework* de otimização *tree-ssa*, que, além de realizar a eliminação de código inútil, introduz funções para auto-vetorização.

Aplicativos científicos podem finalmente ser escritos em Fortran 95, uma vez que o suporte a essa linguagem foi introduzido com a nova versão. Assim, é possível agora compilar os quatro *benchmarks* “Fortran 90” do conjunto de comparativos de desempenho de ponto flutuante SPECfp2000.

Os desenvolvedores da GCC dedicaram atenção especial aos *front ends* para C e C++ na nova versão. Um comparativo de compilação para essas duas linguagens resultou em uma redução de tempo de 25%; além disso, agora as classes aninhadas de modelos podem ser declaradas como *friend*, além de muitas outras coisas.

A “onda” de 64 bits chegou também para a GCC: uma série de funções matemáticas como `asin()`, `acos()` ou `log10()` estão disponíveis como *intrinsics* para as extensões do AMD64, desde que o código fonte seja compilado com a opção `-ffast-math`. Algo semelhante existe também para a plataforma Itanium (IA64), na qual divisões inteiras e de ponto flutuante, bem como o cálculo de raízes quadradas, são realizadas de modo *inline*. Máquinas SPARClike não são mais suportadas, nem o OpenBSD para a plataforma SPARC.

Detalhes sobre as modificações e novidades podem ser encontrados, como sempre, no *changelog* do projeto. O código fonte da GCC 4.0.0 está disponível para download em diversos servidores espelho. ■

<http://gcc.gnu.org/>

<http://gcc.gnu.org/projects/tree-ssa/>

<http://gcc.gnu.org/projects/tree-ssa/vectorization.html>

<http://gcc.gnu.org/fortran/>

<http://www.spec.org/cpu2000/CFP2000/>

<http://gcc.gnu.org/gcc-4.0/changes.html>

<http://gcc.gnu.org/mirrors.html>

Ferramentas para computação virtual no Linux

Hóspede Anfitrião &

Antigamente só se podia escolher um único sistema operacional. Depois tornou-se possível usar mais, mas era preciso reiniciar. No mundo de hoje, os usuários querem ter sua heterogeneidade num único pacote.

Por que não rodar aplicativos do Windows ou o próprio Windows no Linux? Por que não rodar o Mac OS X ou o Linux, dentro de um sistema Linux "hospedeiro"? Os desenvolvedores gostam dos sistemas virtuais porque um aplicativo pode derrubar o PC virtual sem derrubar o sistema real que o abriga. Os administradores de sistemas usam máquinas virtuais para testar versões do kernel ou experimentar novas configurações de rede. Os usuários querem apenas rodar seu aplicativo proprietário favorito. Neste mês, mostraremos o caminho para a computação virtual no Linux.

A API do Wine: O *Wine* e as ferramentas derivadas dele permitem rodar aplicativos do Windows® no Linux. Mostraremos como você pode fazer isso e apresentaremos uma dupla de variantes do Wine: o *CrossOver Office* e o *Cedega*.

VMWare 4.5.2: Uma boa escolha para quem busca uma ferramenta estável e testada que lhe forneça uma máquina virtual capaz de rodar o Windows e outros sistemas dentro do Linux.

Mac-on-Linux: *Mac-on-Linux* é uma ferramenta livre que permite rodar o Mac OS (Clássico ou X) dentro de sistemas Linux em máquinas Power PC. Mostraremos como você pode extrair o melhor dos dois universos sem recorrer ao *dual-booting*.

User-Mode Linux: O User-Mode Linux (UML) é uma ferramenta rápida, racional e barata para rodar um sistema virtual Linux dentro de uma máquina com Linux. Descreveremos como configurar seu próprio Linux virtual e mostraremos porque usuários, desenvolvedores e administradores de sistemas descobriram milhares de usos para o UML.

Xen: Este software, desenvolvido pela Universidade de Cambridge, tem desempenho muito superior à concorrência. Mostraremos como usá-lo para substituir uma rede completa por apenas uma máquina, com óbvias vantagens na redução de custos.

Água ou vinho? 23

Se você precisa rodar um aplicativo do Windows no Linux, não há opção melhor que o Wine. Analisamos suas variantes livres e comerciais, como o *Crossover Office* e *WineX*, para ver como elas se viram em situações da vida real.

PC numa caixa 28

Analisamos a versão 4.5.2 do VMWare, a mais nobre e tradicional entre as máquinas virtuais.

Linux no Linux 32

O *User-mode Linux* se parece com Linux porque é Linux. Existem mil maneiras de usar esse sistema Linux virtual rápido e racional; invente a sua.

Uma maçã por dia 36

Os emuladores agora oferecem aos usuários a capacidade, antes inimaginável, de rodar o Mac OS X no Linux, matando vários coelhos com uma só cajadada.

Multiplicando Pingüins 40

O *Xen* implementa o conceito de virtualização, muito popular no mundo dos mainframes, no Linux, permitindo que você rode vários "sistemas" virtuais individualizados em uma única máquina.



Wine, Crossover Office e Cedega

Água ou Vinho?

Se você precisa fazer com que um aplicativo do Windows rode no Linux, não há opção melhor que o Wine. Investigamos as variantes livres e comerciais do Wine para ver como elas se viram em situações da vida real.

POR BERND VILLIGER, OLIVER FROMMEL E JÖRG LUTHER

Em 1993, a Sun Microsystems desenvolveu uma pequena ferramenta chamada WABI, concebida para permitir que os usuários rodassem diretamente aplicativos do Windows® no Solaris – ou seja, sem a necessidade de portar os aplicativos antes. Apenas alguns meses depois, ocorreu a primeira tentativa de fazer com que a ferramenta rodasse em BSD e Linux; esse passo marca o nascimento daquilo que hoje é conhecido como projeto Wine. O nome *Wine*, um acrônimo na mais típica tradição GNU, significa *Wine is not an Emulator* (O Wine não é um emulador).

E o Wine [1] não é realmente um emulador no sentido tradicional. A ferramenta não emula nenhum tipo diferente de CPU, nem, estritamente falando, um sistema operacional. Em vez disso, o Wine se insere como uma camada de tradução entre o Linux e o aplicativo para Windows® que o usuário deseja rodar. Ele intercepta quaisquer chamadas à API do Windows® feitas pelo aplicativo e tenta convertê-las em chamadas equivalentes de Unix e X11. De um ponto de vista técnico, o Wine é um *loader* (“carregador”), que carrega e roda programas do Windows®, e uma série de bibliotecas que traduzem ou emulam chamadas às funções da API do Windows®.

A maior parte dos problemas do Wine reflete seu papel como tradutor entre os aplicativos para Windows® e o Linux. A excelência de uma tradução depende inteiramente do conhecimento que o tradutor tenha da linguagem e da cultura. Todos sabem que a Microsoft não aprecia que estranhos obtenham conhecimento das operações internas de seus sistemas operacionais e aplicativos. Mas a Microsoft não é a única fabricante de software que gosta de guardar segredos; muitas outras relutaram em portar seus produtos para Linux.

O Acrobat Professional, da Adobe, é um bom exemplo desse fenômeno; eis porque optamos por ele como software de teste. Nossa máquina de testes foi um laptop HP com um processador Pentium 4 de 1.7 GHz e 256 Mbytes de RAM. O sistema operacional escolhido foi o SUSE LINUX 9.0 Professional.

Instalação do Wine

Quase todas as principais distribuições incluem o Wine, embora o pacote não seja normalmente instalado por padrão. Para descobrir se o Wine já está instalado em seu sistema, digite o seguinte comando (que só funciona em distribuições baseadas em RPM):

Quadro 1: Removedor de Wine

Não importa o que você faça, não tente rodar instalações paralelas ou múltiplas do Wine. Antes de instalar uma nova versão, o bom-senso manda remover a anterior.

Em primeiro lugar, descubra qual versão do Wine está instalada:

```
# rpm -qa | grep -i wine
wine-20040505-1
```

Seu sistema deve responder com o rótulo da versão – que você pode apontar como o programa a ser removido. Em nosso exemplo, essa versão é a 20040505-1 e, portanto, o comando para removê-la é `rpm -e wine-20040505-1`. Remova então os diretórios criados pelo Wine para cada conta:

```
# rm -rf /root/.wine
# rm -rf /home/usuário/.wine
[...]
```

Essa etapa remove os depósitos de lixo, abrindo o caminho para um upgrade do Wine. Use o mesmo método para remover uma instalação do Wine que tenha dado errado.

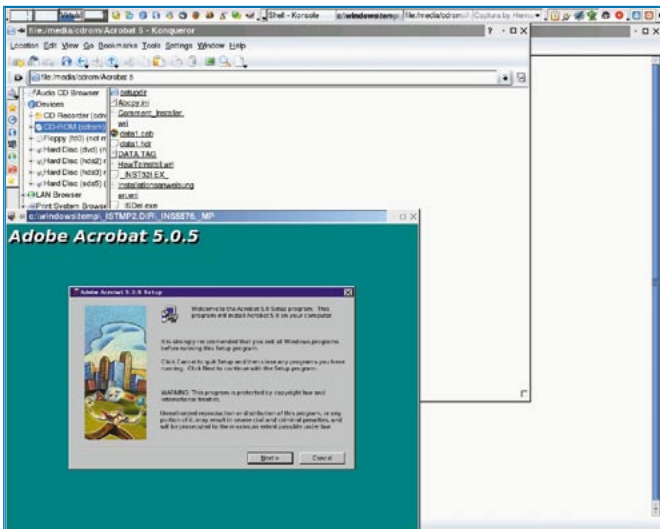


Figura 1: Uma visão estranhíssima: o instalador do Adobe Acrobat para Windows® em execução sob o Linux, graças ao Wine.

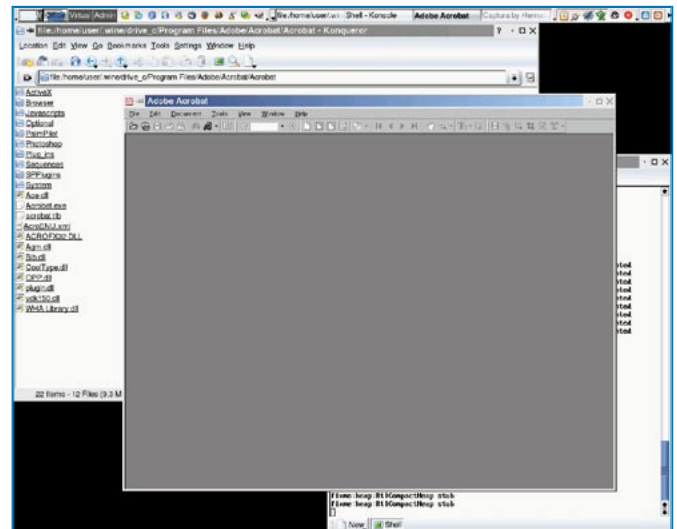


Figura 2: Embora não seja oficialmente suportado pelo Wine, o Adobe Acrobat mostrou-se cooperativo.

```
rpm -qa | grep -i wine
```

Se o Wine estiver instalado, você verá uma mensagem parecida com `wine-20040505-1` (no SUSE 9.0) ou `wine-20040813-7` (no SUSE 9.2). Como você pode ver, a data de lançamento é também o número da versão.

De qualquer modo, talvez você queira substituir quaisquer versões mais antigas que tenha encontrado por uma mais recente (de 2005). Isso é muito fácil e traz a vantagem de uma útil configuração padrão na maioria dos casos. Porém, será necessário desinstalar a versão anterior para evitar problemas (veja o **quadro 1: Removedor de Wine**).

A instalação também não é nada espetacular. Primeiro, baixe o pacote atual (que tem cerca de 12 MB) do quartel-general do Wine [2]. Em seguida entre como root e o instale com o comando:

```
rpm -ivh wine-20050111-SuSELinux90.i586.rpm.
```

Isso de fato completa a configuração em uma tacada só, já que a versão atual vem pré-configurada e pronta pra uso. Se você desejar ver ou modificar a configuração, basta rodar a ferramenta `winecfg`,

que permite visualizar ou alterar as atribuições de letras dos drives virtuais que representam seus dispositivos.

Instalação de aplicativos do Windows

O próximo passo é instalar seus primeiros aplicativos Windows. Para nosso teste, optamos por um aplicativo popular que não é oficialmente suportado pelo Wine: o *Adobe Acrobat 5*.

A fonte de instalação é o CD do programa; o leitor de CD-ROM de nossa máquina de testes está atribuído ao “drive” `Z:`. Assim, para iniciar a instalação basta digitar o comando:

```
wine 'Z:\media\cdrom\Acrobat 5\Setup.exe'
```

Certifique-se de usar nomes de caminhos e drives no estilo Windows®. Use também aspas simples (') ou duplas (") para o caminho do programa. A falta disso faz com que o shell do Linux destroe o comando, os espaços e o texto que se segue.

A instalação do Adobe Acrobat (**figura 1**) levou cerca de 15 minutos em nosso laboratório – e trata-se apenas de um aplicativo simples. Mas a espera valeu a pena: após a conclusão do processo, o comando:

```
wine 'C:\Program Files\Adobe\Acrobat.exe'
```

nos trouxe uma janela de trabalho do Adobe Acrobat, como você vê na **figura 2**.

Admitamos que nem todas as instalações são assim macias. Tentativas de instalar qualquer versão do *NetObjects Fusion* (outro programa que não é oficialmente suportado) demonstra as ciladas típicas. O aplicativo não pode ser instalado sem que antes se instale o *Internet Explorer*.

Alternativa: Crossover Office

Se você estiver realmente interessado em usar o Wine num ambiente de produção, dê uma olhada no pacote Crossover Office da CodeWeavers [3], um derivado comercial do Wine que custa entre US\$ 39.95 (versão standard) e US\$ 69 (versão profissional multiusuário). O Crossover Office oferece uma interface gráfica e um manejo mais amigável. Porém, não remove o problema enfrentado pelo Wine (ver **quadro 2: Wine não é um emulador**). Assim, o Crossover Office Standard é basicamente uma “caixa de areia” que facilita o uso do Wine.

Em primeiro lugar, vejamos como o Crossover lida com uma instalação do Acrobat. Isso nos permitirá testar as qualidades do programa numa cena de vida



Figura 3: Terminada a instalação (bem simples, por sinal) é possível configurar diretamente o Crossover Office.

real. Usamos o pacote de demonstração do Crossover, que funciona por um mês após o download, embora não seja preciso registrar-se para baixá-lo.

O processo de instalação incomum, mas eficiente, baseia-se num script shell:

```
install-crossover-standard-demo-4.1.sh
```

A instalação não tem nada de mais e os resultados são auto-explicativos (**figura 3**). Na maioria dos casos, você pode simplesmente aceitar a configuração padrão. Ao terminar, deve surgir a janela principal, como mostra a **figura 4**.

Encontramos apenas um ponto passível de crítica: para evitar conflitos num estágio posterior, é necessário baixar as fontes tipicamente instaladas pelo Windows. A aba *Fonts* na janela principal faz o trabalho de preparação por você, de forma

que é possível eliminar o problema simplesmente baixando as fontes necessárias.

Novamente, nossos testes começaram com a tentativa de instalar o Adobe Acrobat Professional 5.0. Para isso, apenas clicamos em *Install* na aba *Add/Remove*. Na janela que se abriu, selecionamos *Unsupported Software* e apontamos para a fonte de instalação (que é tipicamente */media/cdrom*).

Depois de instalar um aplicativo, pode-se usar a aba *Menus* na janela principal do Crossover ou a velha e boa linha de comando para iniciar o aplicativo:

```
~/cxoffice/bin/acrobat
```

Não faz muito sentido especificar uma versão do Windows® como ambiente. Os programas do Windows® normalmente rodam melhor na configuração padrão. Os aplicativos que necessitam especificamente do Windows NT 4 ou NT 5 (Windows 2000) normalmente não rodam (ao menos não com facilidade) no Wine, como provou nosso teste com o Adobe Acrobat 6: o instalador (do Windows®) travou imediatamente.

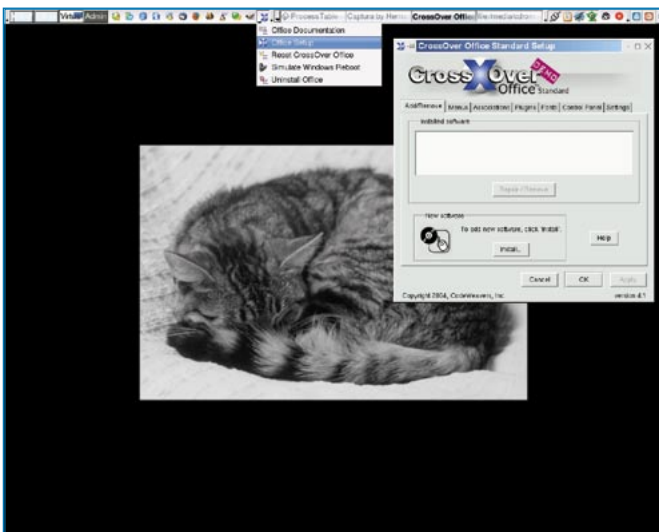


Figura 4: As etapas de instalação e configuração do seu Windows de mentirinha ocorrem na janela principal do Crossover Office.



Figura 5: A janela que simula um “reboot” do Windows traz o ícone de uma bota.

Diferente do Wine, o Crossover Office tem um método mais simples para reiniciar o Windows®. Basta digitar o seguinte comando no terminal:

```
./cxoffice/bin/cxreboot
```

A **figura 5** mostra um exemplo da mensagem que surge na tela como resultado. Claro que, para alguns, ela pode facilmente ser mal-compreendida...

Conclusão

O Wine é uma tentativa muito ambiciosa de rodar aplicativos do Windows nativamente em sistemas Unix. Sendo um aplicativo minimalista, o Wine se restringe a oferecer um serviço de tradução entre o sistema operacional e o aplicativo. A simplicidade do suporte a software e a hardware é uma das principais vantagens do programa.

Quadro 2: O Wine não é um emulador

Diferente do Wine, um verdadeiro emulador cria uma área isolada em que simula um computador. Às vezes é preciso até mesmo instalar o sistema operacional separadamente para usar essa “máquina dentro da máquina”. Baseado no mesmo princípio, você pode instalar e usar qualquer aplicativo escrito para o sistema operacional da máquina virtual. A perda de desempenho devido à sobrecarga da simulação e o preço das licenças (a máquina virtual requer um sistema operacional completo) são os maiores empecilhos desse método.

O Wine implementa uma alternativa lógica: em vez de simular o sistema operacional como um todo, limita-se a forjar um ambiente que faz com que os aplicativos para Windows® se sintam em casa. Isso leva ao nome usado internamente pelo Wine, *fake windows* (windows falso):

```
# ls -a /home/user/.wine
dosdevices fake_windows [...]
```

A estrutura de dados no diretório do Wine baseia-se na estrutura da “coisa real”. Porém, o fac-símile é muito diferente, o que é a maior fraqueza do projeto. Os aplicativos que acessam elementos específicos do Windows® com frequência dão com a cara na parede (virtual).

Quadro 3: Brincando com o Cedega

Se você quiser rodar jogos do Windows® no Linux, precisará do *Cedega*, da Transgaming [4]. O Cedega é uma versão do emulador de API do Wine à qual os desenvolvedores acrescentaram funções da biblioteca multimídia DirectX. O Cedega suporta até mesmo jogos atuais do Windows® com gráficos complexos como Half Life 2 ou Far Cry.

Porém, esse tipo de performance não é de graça: a fabricante, a norte-americana Transgaming, cobra US\$ 5,00 pela assinatura mensal do produto, com um período mínimo de três meses.

Se você quiser apenas dar uma olhada no Cedega, pode baixar e compilar sozinho a maior parte do software; porém, o download carece de alguns componentes fundamentais, o que significa que muitos jogos protegidos contra cópia não vão rodar.

Instale o Cedega a partir do CVS

Sem uma assinatura, os fontes só estão disponíveis através do sistema CVS de controle de versões; é preciso ter o programa `cv`s para baixá-los. Se você tiver o utilitário `cv`s em sua máquina, precisa primeiro iniciar uma sessão no servidor:

```
cv
```

s -d:pserver:cvs@cvs.transgaming.org:~
 /cvsroot login

Se lhe pedirem uma senha, basta digitar `cv`s. Se funcionar, o programa não dará nenhuma resposta; caso contrário, ele vai reclamar. Em seguida baixe os arquivos para sua máquina:

```
cv
```

s -z3 -d:pserver:cvs@cvs.transgaming.org:~
 org:cvsroot co winex

Em pouco tempo, você terá o código fonte completo no diretório `winex`. A compilação segue os passos normais, embora o Fedora Core 3 seja uma exceção à regra geral:

```
./configure --enable-opengl --enable-pthreads
```

O comando `make` inicia o compilador e `su -c 'make install'` coloca o programa nos diretórios corretos no disco. Então você pode copiar o arquivo exemplo de configuração para o lugar certo:

```
cp documentation/samples/
  config ~/.wine/config
```

Antes de começar, é preciso criar alguns arquivos e diretórios. O Cedega espera que parte da estrutura de diretórios se assemelhe ao Windows®.

O arquivo de exemplo pressupõe que você tenha um diretório chamado `~/.transgaming`. Crie um diretório com esse nome ou mude-o para `~/.wine/config`; se o fizer, use esse diretório para as etapas seguintes. Adicione os subdiretórios requeridos como segue:

```
mkdir -p ~/.transgaming/c_drive
  /windows/system32
  mkdir ~/.transgaming/c_drive/windows
  /fonts
```

Talvez você precise acrescentar uma entrada para o drive de CD-ROM ao arquivo de configuração. A entrada para o primeiro drive IDE é assim:

```
[Drive H]
  "Path" = "/media/cdrom"
  "Type" = "cdrom"
  "Label" = "CD-ROM"
  "Filesystem" = "win95"
  "Device" = "/dev/hdc"
```



Figura 6: O jogo Moorhuhn rodando no Cedega.

Agora já dá para instalar um jogo do Windows®. Usamos a edição de inverno do jogo *Moorhuhn*, muito popular na Alemanha e em nosso laboratório. Com o Cedega, é preciso rodar `wine` para instalar o jogo:

```
wine moorhuhn_we.exe
```

Complete as etapas necessárias para colocar o *Moorhuhn* em seu “disco” Cedega, ou seja, seu diretório `.transgaming/c_drive`, que é equivalente ao `C:` do Windows®.

Para ter certeza de que o jogo encontrará seus arquivos, mude para o diretório de instalação e rode-o a partir de lá:

```
cd .transgaming/c_drive/Program Files/Phenimedia AG/Moorhuhn
  Winter-Edition
  wine MoorhuhnWinter.exe
```

O jogo roda sem quaisquer obstáculos e em velocidade normal (ver figura 6). Os outros jogos que testamos instalaram no Cedega, mas emitiram mensagens de erro e travaram quando tentamos rodá-los, porque todos suspeitaram que o Cedega estava tentando solapar o esquema de proteção contra cópia.

A versão comercial do Cedega tem código que evita que isso aconteça. Se você quiser rodar jogos atuais protegidos contra cópia, não há outra alternativa senão a versão comercial do programa.

O CrossOver Office ajuda usuários não-técnicos do Linux a ganhar acesso ao poder do Wine. Mas antes de comprar o Crossover Office, faça um teste com a versão *Trial*, ou veja a lista de compatibilidade no site oficial [3], para se certificar de que o aplicativo para Windows de que você precisa realmente roda no Linux. Entre os compatíveis estão softwares po-

pulares como o Microsoft Office, Microsoft Internet Explorer, Microsoft Outlook, algumas versões do Adobe Photoshop e o Macromedia Dreamweaver, entre muitos outros. Entretanto, se um aplicativo se recusar a rodar no Wine ou no Crossover Office, sua única alternativa será usar uma máquina virtual, como *VMware* ou a alternativa livre, o *Qemu*. ■

INFORMAÇÕES

[1] WineHQ: <http://www.winehq.org/>

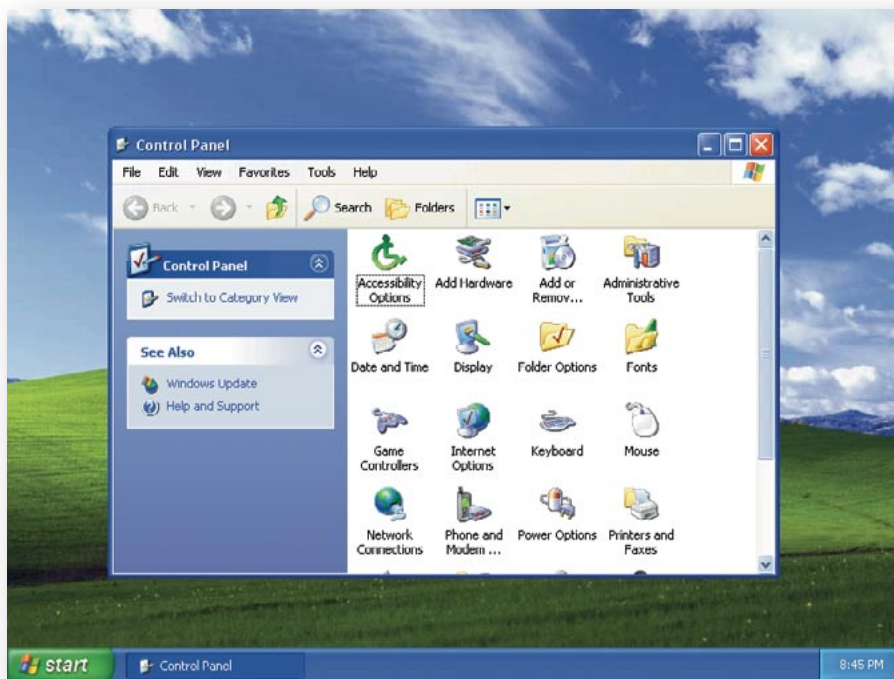
[2] Pacotes do Wine: <http://www.winehq.org/site/download>

[3] Codeweavers: <http://www.codeweavers.com/>

[4] Transgaming: <http://www.transgaming.com/>

VMware Workstation 4.5.2 para Linux

PC numa caixa



Atualmente em sua versão 4.5.2, o VMWare é, para muitos usuários, o mais nobre e tradicional dos PCs virtuais.

POR HAGEN HÖPFNER

Apesar do sucesso dos aplicativos de compatibilidade baseados no projeto *Wine*, talvez você ainda precise emular um PC completo. Por exemplo, se você tiver que testar um programa para Windows que não funciona no *Wine*, configurar algumas máquinas de teste sem ser restringido por tabelas de partição ou apenas testar seu novo website usando uma variedade de sistemas operacionais e navegadores, uma máquina virtual é a solução.

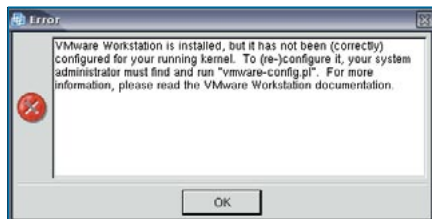


Figura 1: Essa mensagem de erro é exibida se você tentar iniciar o VMWare sem configurá-lo após terminar a instalação.

O tolo e o falsário

Em contraste com “tradutores” de API como o *Wine*, o *VMware Workstation* é um software que emula um computador completo. O chamado “sistema operacional convidado” roda dentro da simulação do VMware e não pode distinguir esse ambiente de um PC real. A VMware [1], a empresa por trás do produto de mesmo nome (que se tornou subsidiária da EMC Corporation em 2004), tem variantes para Linux e para Windows® de seu emulador. A versão da caixinha custa R\$ 500,00 na loja da VMware [2]; a versão para download custa apenas 10 dólares a menos [3].

Para que a emulação rode numa velocidade aceitável, o PC hospedeiro precisa ser bastante rápido. A VMware recomenda um Pentium II ou processador AMD equivalente, 256 MB de RAM, um adaptador de vídeo de 16-bits e ao menos 20 MB de

espaço em disco livre para o software. É preciso contar espaço a mais no disco para que o hospedeiro o conceda ao sistema convidado, além de uma placa de rede – se o PC emulado precisar acessar a rede.

Essas especificações devem ser consideradas a configuração mínima. O VMware permite o uso de quase qualquer sistema operacional da Microsoft como hóspede, desde o MS DOS até o Windows Server 2003, além da maioria das principais distribuições Linux, do Novell Netware 5 e 6, Solaris 9 e 10 para X86 e FreeBSD. Extra-oficialmente, porém, o VMware suporta uma ampla gama de outros sistemas operacionais. O emulador engana o sistema operacional convidado para que detecte o hardware listado no **quadro 1: Hardware emulado**.

O sistema de nosso laboratório para a elaboração deste artigo usou o VMware para Linux versão 4.5.2 num PC anfi-

trião com SUSE Linux 9.1, um processador AMD Athlon XP 1900+, 512 MB de RAM, um adaptador de vídeo NVIDIA GeForce com o chipset MX4000 (com driver binário da própria NVIDIA) e um adaptador WLAN.

Aquisição do VMware

Como mencionado anteriormente, uma licença do VMware não é exatamente barata. Se você preferir testar o emulador antes de investir 199 dólares americanos, pode aproveitar o período de teste de 30 dias. A VMware simplesmente pede que você se registre antes de baixar o programa [3] e manda um email com uma senha de teste. Usuários registrados podem então baixar um pacote *rpm* e um arquivo *tar*. Em nosso laboratório, usamos o pacote RPM, que é bastante fácil de instalar com o comando `su -c "rpm -Uvh VMware-workstation-4.5.2-8848.i386.rpm"`.

A tentativa de iniciar o programa digitando `vmware` logo depois da instalação provoca a mensagem de erro mostrada na **figura 1**. É preciso antes rodar o script de configuração `vmware-config.pl`. Este apresenta o hospedeiro ao emulador. O PC virtual acessa o hardware físico por partes e necessita de seus próprios módulos do kernel no sistema hospedeiro para fazê-lo.

O VMware não tem um módulo pré-compilado do kernel para o SUSE Linux 9.1; em vez disso, o script de configuração gera o módulo. Para fazer isso, ele necessita de um compilador C, tipicamente o `gcc` no Linux. No SUSE 9.1 o compilador está em `/usr/bin/gcc`, como você pode descobrir digitando `whereis gcc`.

Bebendo na fonte

O módulo VMware não será compilado a menos que você tenha os fontes do kernel atual. O SUSE guarda os fontes no pacote *kernel-source*, que você provavelmente terá de instalar e modificar para combinar com seu kernel. Isso é feito facilmente

Quadro 1: Hardware emulado

O VMware emula um PC completo com uma plataforma fixa de hardware. O hardware emulado nada tem a ver com o equipamento real que há no computador hospedeiro; o sistema hóspede vê apenas os componentes virtuais.

Processador:

- ⇒ Intel Pentium II ou mais recente, AMD Athlon ou mais recente (depende da CPU original)
- ⇒ Suporte experimental a CPU AMD64 e IA 32e

RAM:

- ⇒ Até 3.6 GB dependendo da memória física
- ⇒ Máximo de 4 GB para todas as máquinas virtuais

Drives IDE:

- ⇒ Até quatro dispositivos IDE
- ⇒ Discos rígidos virtuais (máx. 128 GB) ou físicos
- ⇒ Drives de CD/DVD-ROM

Dispositivos SCSI:

- ⇒ Até sete dispositivos
- ⇒ Discos rígidos virtuais (máx. 256 GB) ou físicos
- ⇒ Suporte a SCSI genérico
- ⇒ Suporte a scanners, CD/DVD ROMs, drives de fita etc.
- ⇒ Controlador LSI Logic LSI53C1030 Ultra320 SCSI I/O
- ⇒ Adaptador Mylex (BusLogic) BT-958 (apenas no Windows XP, com drivers adicionais)

Mídia removível:

- ⇒ Drives de CD-ROM/CD-R/CD-RW (físico ou imagem ISO)
- ⇒ Drive de DVD-ROM
- ⇒ Máximo de duas unidades de disco de 1.44 MB (físico ou imagens)

Placas de vídeo:

- ⇒ VGA e SVGA com VESA BIOS

Portas:

- ⇒ Máximo de quatro portas seriais (COM)
- ⇒ Máximo de duas portas paralelas bidirecionais (LPT)
- ⇒ Dois controladores USB 1.1 UHCI
- ⇒ Teclado Windows de 104 teclas
- ⇒ Mouse PS/2

Dispositivos de rede:

- ⇒ Máximo de três adaptadores de rede virtuais (Ethernet)
- ⇒ Placa de rede compatível com AMD PCnet PCI II

Placa de som:

- ⇒ Permite gravação e reprodução
- ⇒ Compatível com Creative Sound Blaster Audio PCI (não há suporte a MIDI ou Joysticks)

BIOS:

- ⇒ Suporte a PhoenixBIOS 4.0 Release 6 VESA BIOS com DMI-v2.2/SMBIOS

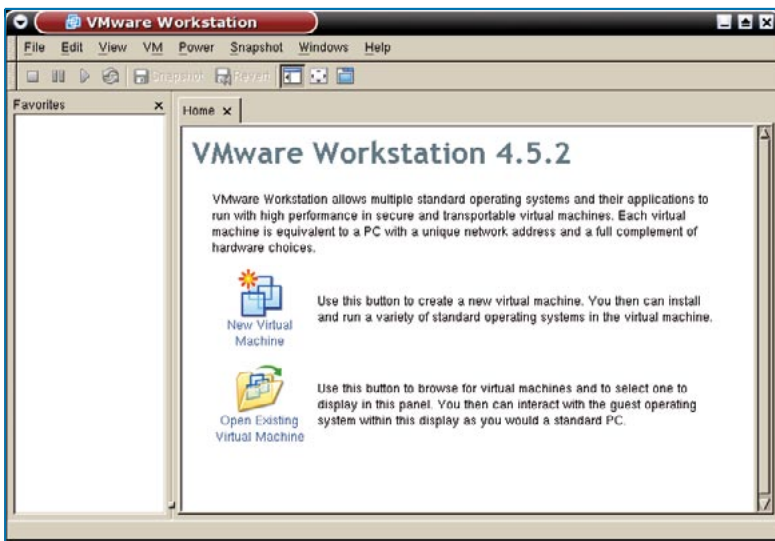


Figura 2: Quando iniciado pela primeira vez, o VMware não está configurado. Você pode criar uma nova máquina virtual ou carregar uma configuração existente.

com uma pequena ajuda do YaST e logo você pode seguir em frente e preparar os fontes:

```
cd /usr/src/linux
su -c "make cloneconfig && make prepare-all"
```

Após completar essas etapas, digite `su -c vmware-config.pl` para iniciar o script de configuração. Esse script manda que você leia e aceite o contrato de licença. Quando lhe perguntarem onde estão guardados os arquivos de cabeçalho do kernel (*kernel headers*), basta pressionar **[Enter]** para aceitar a configuração padrão do script. Após um tempinho compilando o binário, você precisa informar se necessita de uma versão do VMware que trabalhe

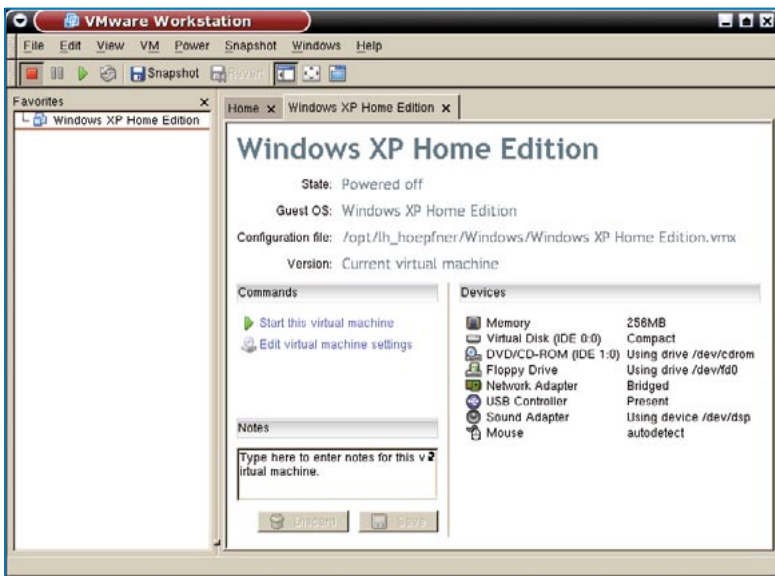


Figura 3: O usuário decidiu instalar o Windows XP na máquina virtual.

Quadro 2: Distribuições hospedeiras

A versão para Linux do VMware roda em várias das principais distribuições Linux. Como ele vem com seu próprio módulo do kernel, o uso do VMware em quaisquer outras variantes do Linux é bem incômodo. O fabricante permite o uso das seguintes distribuições:

- ⇒ Mandrake Linux 9.0 (Kernel 2.4.19)
- ⇒ Mandrake Linux 8.2 (Kernel 2.4.18-6mdk)
- ⇒ Red Hat Enterprise Linux 3.0 (Kernel 2.4.21 or 2.4.21-15.EL)
- ⇒ Red Hat Enterprise Linux 2.1 (Kernel 2.4.9-e3)
- ⇒ Red Hat Linux Advanced Server 2.1 (Kernel 2.4.9-e3)
- ⇒ Red Hat Linux 9.0 (Kernel 2.4.20-8 or 2.4.20-20.9)
- ⇒ Red Hat Linux 8.0 (Kernel 2.4.18)
- ⇒ Red Hat Linux 7.3 (Kernel 2.4.18)
- ⇒ Red Hat Linux 7.2 (Kernel 2.4.7-10, 2.4.9-7, 2.4.9-13, 2.4.9-21 or 2.4.9-31)
- ⇒ Red Hat Linux 7.1 (Kernel 2.4.2-2 or 2.4.3-12)
- ⇒ Red Hat Linux 7.0 (Kernel 2.2.16-22 or 2.2.17-14)
- ⇒ SuSE Linux 9.1 (Kernel 2.6.4-52)
- ⇒ SuSE Linux 9.0 (Kernel 2.4.21-99)
- ⇒ SuSE Linux Enterprise Server 8 (Kernel 2.4.19)
- ⇒ SuSE Linux 8.2 (Kernel 2.4.20)
- ⇒ SuSE Linux 8.1 (Kernel 2.4.19)
- ⇒ SuSE Linux 8.0 (Kernel 2.4.18)
- ⇒ SuSE Linux Enterprise Server 7 (Kernel 2.4.7 and patch 2)
- ⇒ SuSE Linux 7.3 (Kernel 2.4.10)

em rede (*network-capable*). Se for o caso, você deve optar também por suporte a NAT e permitir que o script atribua uma sub-rede privada.

Compartilhando uma rede e um disco

Responda “sim” à pergunta *Do you want to be able to use host-only networking in your virtual machines?* (você quer usar redes *host-only* em sua máquina virtual?), se você quiser que essa variante de rede esteja disponível mais tarde. *Redes host-only* significa que o PC emulado só pode acessar o sistema hospedeiro através da rede, e não diretamente pelo hardware. O PC emulado pode usar a conexão à Internet do hospedeiro em NAT ou *bridge mode*. Em *bridge mode*, o PC emulado se comporta como se estivesse ligado diretamente à LAN. Precisa de um endereço IP próprio, que deve ser válido para a rede local.

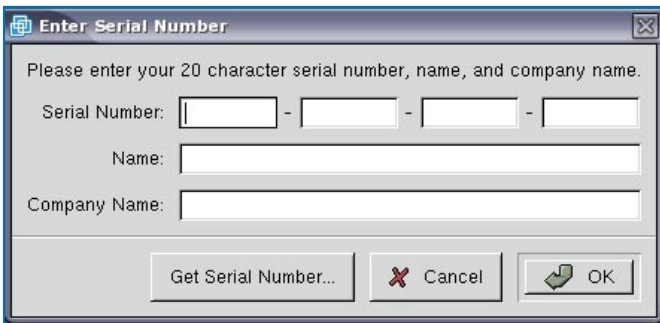


Figura 4: O VMware não iniciará a menos que você digite um número de série válido. Você recebe um ao adquirir uma licença ou ao baixar a versão de teste de 30 dias.

Se você quer que o PC emulado monte o sistema de arquivos do hospedeiro, o script cria um dispositivo de rede *host-only*. Então o VMware usa Samba para acessar o hospedeiro. O script pede apenas um nome de usuário e uma senha para configurar o Samba – porém, usuários do Samba devem ser usuários regulares no hospedeiro. Para acrescentar usuários posteriormente, torne-se root e digite `/usr/bin/vmware-smbpasswd vmnet1 -a username. vmnet1` é o dispositivo de rede do Samba.

Primeiros passos

Após terminar a configuração, o VMware vai iniciar como mostrado na **figura 2**. Agora você pode clicar no botão *New Virtual Machine* para iniciar o assistente de configuração. Ele pede que você escolha o sistema operacional hóspede que será instalado, que especifique um arquivo para abrigar o sistema de arquivos emulado (um “HD virtual”) e que decida o tamanho do arquivo. Os resultados são mostrados na **figura 3**.

Clique no botão *Start this virtual machine* para iniciar seu novo PC virtual. Nesse momento a máquina emulada ainda não tem um sistema operacional – é preciso instalá-lo, exatamente como você faria num computador real. Para isso, insira o CD de instalação do sistema operacional a ser usado no PC hóspede no drive de CD-ROM do hospedeiro.

janela mostrada na **figura 4**; Na dúvida, o item do menu *Help | Enter Serial Number* leva você ao lugar certo.

Instalação do Hóspede

Para instalar um sistema operacional direto do CD, é preciso habilitar suporte a *Legacy* para o CD ROM com a versão de teste. O item *DVD/CD-ROM*, que permite configurar essa opção, está no menu *Devices* do PC virtual.

Se o programa de instalação pede que você forneça dados, será preciso antes certificar-se de que o PC emulado está realmente usando seu teclado e seu mouse. Para fazê-lo, clique na janela do emulador. O VMware sequestrará o teclado e o mouse nesse momento; mas você pode pressionar **[Ctrl]** e **[Alt]** simultaneamente para tirá-los da prisão.

Se você estiver instalando o Windows® ou Linux como sistema operacional hóspede, é uma boa idéia instalar as ferramentas do VMware (*VMware Tools*) no hóspede [4]. Essas ferramentas aumentam o desempenho da placa de vídeo virtual, permitem maior resolução e dizem ao VMware que deixe o mouse sair de cena quando você o empurra para fora dos limites da janela. Isso evita que você tenha que repetidamente pressionar **[Ctrl]** e **[Alt]**.

O item *VM | Install VMware Tools...* diz ao VMware que carregue uma imagem de CD no drive de CD emulado. Os drivers necessários estão nesse “falso” CD.

Antes de iniciar, a versão não-registrada do VMware avisa que você precisa de um número de série. Se você adquirir a licença, ela inclui um número de série; o número da versão de teste chega a você via email. Digite o número na

Cuidado com novas distribuições

Em nosso laboratório de testes, não conseguimos fazer com que o VMware rodasse adequadamente no SUSE LINUX 9.2. Embora as fases de instalação e configuração tenham se completado como descrito neste artigo, o VMware simplesmente se recusava a sair para brincar. O sistema hospedeiro foi incapaz de configurar seu adaptador USB WLAN e só conseguimos iniciar o PC virtual a partir do CD por pura sorte.

O VMware, de fato, informa ao usuário que aquele sistema é novo demais. Antes de começar, dê uma olhada na lista de distribuições Linux oficialmente suportadas (ver **quadro 2: Distribuições hospedeiras**) e leve essa lista a sério. Se você realmente quiser usar uma distribuição Linux que não esteja na lista, antes de qualquer coisa baixe a versão de testes do VMware ou procure por atualizações do programa, para se certificar de que ele vai mesmo funcionar. ■

SOBRE O AUTOR

Hagen Höpfner é Doutor em Ciências da Computação pela Universidade Internacional de Bruchsal, na Alemanha. Já tocou guitarra com amigos na banda “Gute Frage” e em seu tempo livre gosta de fuçar em sua coleção de computadores, todos rodando Linux.



INFORMAÇÕES

- [1] Site oficial do VMware: www.vmware.com
- [2] Loja da VMware: www.vmware.com/vmwarestore/newstore
- [3] Página de download: www.vmware.com/download/workstation.html
- [4] Ferramentas do VMware: www.vmware.com/support/ws4/doc/new_guest_tools_ws.html
- [5] QEMU, uma alternativa Open Source: fabrice.bellard.free.fr/qemu/

Iniciação ao User-mode Linux

Linux no Linux

O User-mode Linux (Linux no espaço do usuário) se parece com Linux porque é Linux. Existem mil maneiras de usar esse sistema Linux virtual rápido e racional; invente a sua.

POR FABRIZIO CIACCHI

O popular e versátil *User-mode Linux* (UML) [1] cria um sistema Linux virtual totalmente operacional numa máquina anfitriã com Linux. O UML tem muitos usos. Os desenvolvedores de programas “perigosos” o empregam para testar seus aplicativos apropriadamente sem pôr em risco o sistema anfitrião. Usuários comuns de Linux usam o UML para fazer experiências com versões do kernel sem ter de se preocupar com um patch novo ou não testado. Os administradores de sistema usam o UML para testar configurações alternativas sem tirar a rede da empresa do ar. É possível até mesmo rodar múltiplas versões do UML na mesma máquina para simular uma rede completa.

O que é User-mode Linux?

O User-mode Linux não é realmente um emulador nem uma API como o *Wine*. O melhor jeito de explicar o User-mode Linux é começar dando uma olhada no papel do kernel.

Cada aplicativo ou programa que você roda dispara um ou mais processos – pense num processo como sendo uma “tarefa” que o programa desempenha. O kernel tem a função

de executar e administrar os processos de todos os programas e “falar” com o hardware. Quando um processo quer se comunicar com um dispositivo (por exemplo, para exibir algo no monitor, enviar dados pela rede, imprimir um documento ou copiar um arquivo

Escolha de emulação

Talvez o melhor modo de compreender as vantagens do UML seja considerar que software desse tipo vem em três formas

- ⇒ Emulação de software
- ⇒ Emulação de hardware
- ⇒ Sem emulação

O Bochs [10] é um dos mais famosos programas emuladores por software. A principal atividade do Bochs é prover a emulação de uma arquitetura de hardware em particular (IA-32, também chamada x86) em cima de um sistema operacional em particular, como Windows®, Mac OS e, obviamente, Linux. Uma vez que o hardware é emulado, é possível instalar qualquer sistema operacional x86 nele (Linux, Windows®, DOS e assim por diante), mas a execução é muito lenta, já que cada instrução do computador precisa ser traduzida do sistema operacional convidado para o sistema anfitrião.

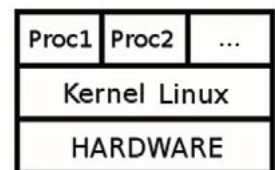


Figura 1: Estrutura de processos normal do Linux.

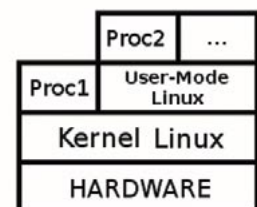


Figura 2: O UML roda como um processo. Nesta imagem, Proc1 está rodando no sistema Linux hospedeiro. Proc2 está rodando no sistema virtual User-mode Linux.

para um disquete), ele pede ao kernel Linux que administre a comunicação com o hardware (figura 1).

O kernel User-mode Linux roda no Linux como um processo do kernel anfitrião. A diferença entre um kernel UML e um kernel comum é que o kernel UML não se comunica diretamente com o hardware. Em vez disso, essa comunicação passa para o kernel “real” da máquina anfitriã, que gerencia a comunicação com o hardware (veja a figura 2).

Uma vez que o sistema virtual e o sistema anfitrião são, ambos, sistemas Linux com estrutura quase idêntica, a comunicação passa de forma muito eficiente do sistema virtual para o anfitrião, demandando pouquíssimo código extra para abstração ou tradução.

A emulação de hardware consiste na reprodução (ou virtualização) da arquitetura nativa de hardware. Esses emuladores são mais eficientes que os emuladores de

software porque usam a infra-estrutura do próprio hardware da máquina hospedeira, mas precisam interceptar todas as chamadas a ele vindas do sistema operacional convidado. Essa solução tem a grande desvantagem de que o código deve ser especializado para uma arquitetura de hardware em particular, que é a mesma para o ambiente do anfitrião e o do convidado. Um exemplo desse tipo de emulador é o *VMware* [2], um programa comercial bastante popular.

O User-mode Linux encaixa-se na última categoria. Ele não precisa emular nenhum hardware específico; em vez disso, fala diretamente com o hardware real. As instruções são passadas com eficácia do kernel UML para o kernel anfitrião. O UML pode executar código nativo e roda, na pior das hipóteses, com uma perda de desempenho de apenas 20% em comparação com o mesmo código sendo rodado no anfitrião.

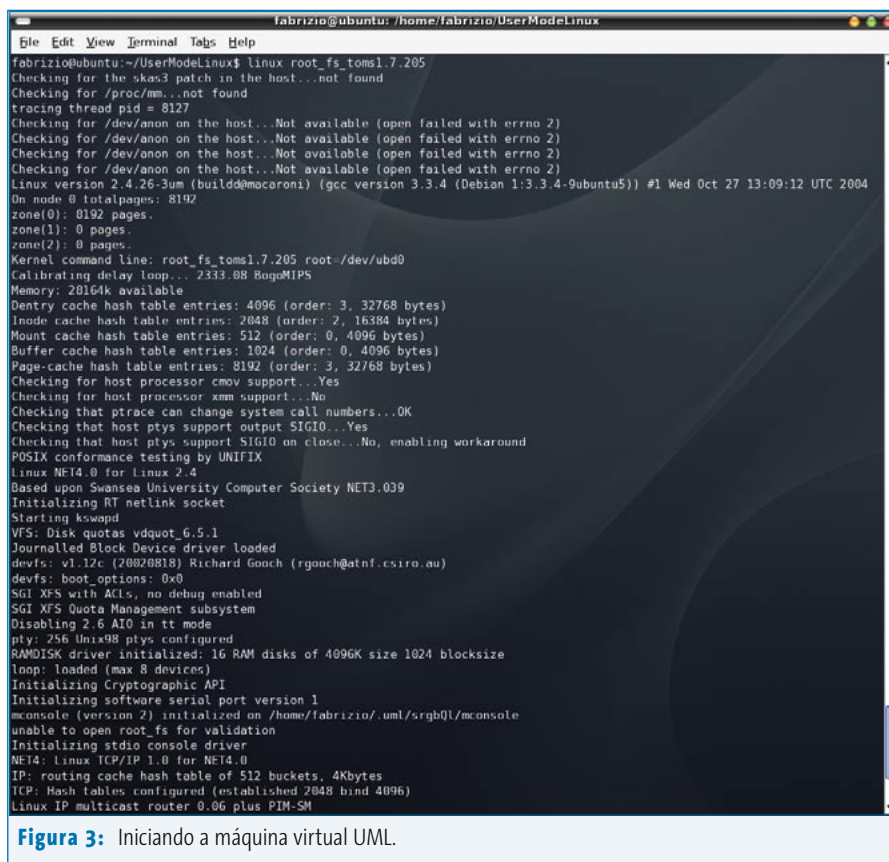


Figura 3: Iniciando a máquina virtual UML.

QEMU: Uma boa alternativa

Se o seu objetivo é usar UML para testar novas distribuições Linux, talvez seja melhor optar pelo emulador de sistema QEMU [10]. Baseado no Bochs [11], o QEMU é muito simples de instalar, configurar e usar. Para saber mais sobre ele, consulte o artigo “Máquinas virtuais: Emulação de sistemas com o QEMU” na edição 8 da Linux Magazine Brasil. Você pode baixar o artigo em PDF de nosso site [12]

Configuração do UML

Instale o User-mode Linux com seu gerenciador de pacotes. Por exemplo, no Debian é preciso digitar o seguinte comando, como root:

```
# apt-get install user-mode-linux \
uml-utilities kernel-patch-uml
```

Esse comando instala o kernel do UML e outros utilitários. Outros gerenciadores de pacotes são igualmente simples, mas se você tiver algum problema ao instalar com um sistema de pacotes ou se tiver problemas de memória durante o boot [3], pode querer baixar um kernel normal (recomendamos a versão 2.4.27 [4]) e o patch para o kernel do UML [5]. Outros patches podem ser encontrados em [6]. Após baixar os arquivos (no mesmo diretório, é claro), abra um terminal e execute os seguintes comandos:

```
$ tar -jxvf linux-2.4.27.tar.bz2
$ bunzip2 uml-patch-2.4.27-1.bz2
$ patch -p1 -d linux-2.4.27 < \
uml-patch-2.4.27-1
$ cd linux-2.4.27
$ make menuconfig ARCH=um
$ make linux ARCH=um
$ strip linux
```

SELinux no UML

Há um documento que explica como configurar um sistema UML com o subsistema de segurança SELinux [13]. Um sistema UML com SELinux pode ser muito útil para criar servidores mais seguros e testar a política de segurança do SELinux sem pôr em risco um sistema “de verdade”.

Depois de completar esses comandos, você terá um arquivo chamado `linux` em seu diretório original. Esse arquivo é o kernel User-mode Linux, que será usado para iniciar o sistema Linux virtual.

Para fazer com que o UML funcione a contento, é necessário incluir duas outras peças do quebra-cabeças: um sistema de arquivos raiz (uma imagem comprimida de uma partição Linux que contenha todos os programas) e os utilitários do UML. Para o sistema de arquivos raiz, todas as imagens disponíveis podem ser encontradas em [7]. Você pode baixar os utilitários do UML em [8] e digitar os seguintes comandos:

```
$ tar -jxvf uml_utilities_XXXXXXX.tar.bz2
$ cd tools
$ make all
$ make install DESTDIR=/
```

XXXXXXX é número da versão das *UML utilities*. Agora você tem um diretório que contém o sistema de arquivos raiz. Lembre-se de pôr o programa `linux` num local que permita usá-lo (se você não o moveu, ele deve estar no diretório `linux-2.4.27`). Digite os seguintes comandos para ler o sistema de arquivos raiz:

```
$ bunzip2 root_fs_toms1.7.205.bz2
$ linux ubd0=root_fs_toms1.7.205
```

O parâmetro `ubd0=` diz à máquina virtual que use o arquivo especificado como sistema de arquivos raiz.

Se tudo correr bem, você verá a máquina virtual iniciando (figura 3) e poderá iniciar uma “sessão virtual” com o nome de usuário `root` e a senha `root`.

Compartilhando o sistema de arquivos raiz.

É possível iniciar duas ou mais máquinas virtuais usando o mesmo sistema de arquivos raiz. O driver `udb0` usa um mecanismo chamado *Copy-On-Write* (COW),

que lê o sistema de arquivos raiz como um dispositivo somente para leitura e armazena as mudanças num arquivo privado de leitura/escrita (o arquivo COW). Se quiser iniciar duas máquinas virtuais (VM1 e VM2) com o mesmo sistema de arquivos, abra duas sessões de terminal e digite os seguintes comandos:

```
[terminal] 1]$ linux ubd0=uml_vm1.cow,
root_fs_toms1.7.205
[terminal] 2]$ linux ubd0=uml_vm2.cow,
root_fs_toms1.7.205
```

Todas as modificações nos dois sistemas virtuais serão escritas nos arquivos COW respectivos. Na verdade, o sistema de arquivos não é compartilhado, mas as duas execuções são independentes. A coisa mais importante a evitar quando os arquivos COW são criados é iniciar o sistema de arquivos diretamente (com `ubd0=root_fs_XXX`), porque todos os arquivos COW registram o tamanho e o *timestamp* (instante em que é montado) do sistema de arquivos raiz e qualquer modificação os tornará inutilizáveis. A sintaxe correta para o reinício seguinte, quando houver um arquivo COW, é esta:

```
[terminal] 1]$ linux ubd0=uml_vm1.cow
[terminal] 2]$ linux ubd0=uml_vm2.cow
```

Redes reais e virtuais

O UML oferece diversas opções interessantes para trabalhar com redes em sistemas virtuais Linux. Assim que você

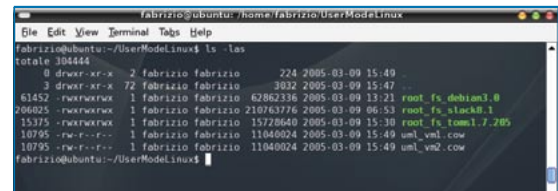


Figura 4: Os arquivos COW das duas máquinas virtuais UML.

tiver seu sistema virtual UML pronto e funcional, pode ter vontade de ligar em rede o sistema virtual com seu hospedeiro ou com outros sistemas virtuais. Há uma descrição detalhada das redes no UML em [9].

A idéia básica por trás das redes no UML é que diversos *transportes* opcionais são oferecidos para administrar a troca de pacotes entre o sistema virtual e o anfitrião. A tabela 1 mostra alguns dos tipos de transporte disponíveis para o UML.

Para habilitar um dispositivo de rede na máquina virtual, digite o texto a seguir na linha de comando do kernel:

```
eth<n>=<transport>,<transport args>
```

onde `<n>` representa a interface real do anfitrião (ou seja, `eth0`) à qual a máquina virtual será conectada. A explicação teórica é que, na máquina virtual UML, há um dispositivo `eth0`. Esse dispositivo é, na verdade, um “apelido” para o dispositivo `tap0` no hospedeiro – trocando em miúdos, o `tap0` do hospedeiro e o `eth0` da máquina virtual UML são o mesmíssimo dispositivo. A interface `tap0` (que, lembre-se, está no sistema real) está diretamente conectada à interface `eth0` do anfitrião. Na prática, isso significa que a `eth0` do

Tabela1: Tipos de transportes UML

Etherap, TUN/TAP	Transportes usados para troca de pacotes entre o sistema virtual e o hospedeiro real.
Switch daemon	Transporte projetado para redes puramente virtuais com outros sistemas UML.
Multicast	Outro transporte projetado para redes virtuais.
Slip, slirp	Transportes usados principalmente quando Ethertap e TUN/TAP não estão disponíveis ou se você não tiver acesso de root à configuração de rede do anfitrião.
Pcap	Transporte que oferece uma interface de rede de somente leitura sendo, portanto, uma boa opção para monitoramento de rede.

sistema virtual “falará” com o mundo pela `eth0` do sistema real, embora ambas possam ter números IP diferentes.

Assim, podemos usar o comando:

```
linux ubd0=root_fs_slack8.1 eth0=ethertap,tap0,fe:fd:0:0:0:1,192.168.0.254
```

para permitir que o UML configure a interface `eth0` da máquina virtual com seu próprio endereço IP. O endereço IP do dispositivo `tap0` real e do `eth0` virtual pode ser o mesmo para configurações mais simples (veja a referência [9] para saber mais sobre configurações de rede mais complexas).

Você precisa então configurar a interface de rede da máquina virtual, através dos arquivos `/etc/hosts`, `/etc/resolv.conf`, `/etc/network`, etc. (exatamente como numa máquina real) para ter acesso à Internet plenamente funcional dentro do ambiente UML.

Conclusão

O User-mode Linux oferece um modo rápido e conveniente de criar sistemas virtuais no Linux. O UML pode ser usado como uma ferramenta para planejar, modelar, testar e resolver problemas em sistemas Linux. O UML também é a base de muitos outros projetos e experimentos, assim como aplicações de negócios e serviços de hospedagem personalizados. Talvez o UML não seja lá muito fácil de instalar e configurar, mas se você conseguir fazer com que funcione, descobrirá muitos usos para ele. ■

SOBRE O AUTOR

Fabrizio Ciacchi (fabrizio.ciacchi.it) – fabrizio@ciacchi.it estuda Ciência da Computação na Universidade de Pisa. Suas atividades principais são estudar Linux, desenvolver sites em PHP e programar em Java. Trabalha também como consultor para diversas empresas e escreve artigos sobre Linux.

INFORMAÇÕES

- | |
|--|
| [1] Site oficial do User-mode Linux: user-mode-linux.sourceforge.net |
| [2] Página oficial do VMWare: www.vmware.com |
| [3] UML em máquinas 2G/2G: user-mode-linux.sourceforge.net/UserModelLinux-HOWTO-4.html#2G-2G |
| [4] Kernel oficial do Linux versão 2.4.27: ftp://ftp.ca.kernel.org/linux/kernel/v2.4/linux-2.4.27.tar.bz2 |
| [5] Patch do UML para o kernel 2.4.27: prdownloads.sourceforge.net/user-mode-linux/uml-patch-2.4.27-1.bz2 |
| [6] Downloads do UML: user-mode-linux.sourceforge.net/dl-sf.html |
| [7] Arquivos de imagem com sistemas raiz: user-mode-linux.sourceforge.net/dl-jails-sf.html |
| [8] Utilitários do UML: prdownloads.sourceforge.net/user-mode-linux/uml_utilities_20040406.tar.bz2 |
| [9] Configuração de rede no UML: user-mode-linux.sourceforge.net/networking.html |
| [10] Site oficial do Bochs: bochs.sourceforge.net |
| [11] Site oficial do QEMU: fabrice.bellard.free.fr/qemu/ |
| [12] Artigo “Benefícios Virtuais” sobre o QEMU (em inglês): www.linux-magazine.com/issue/52/QEMU_System_Emulation.pdf |
| [13] SELinux e UML: www.golden-gryphon.com/software/security/selinux-uml.xhtml |
| [14] Compilação do kernel: user-mode-linux.sourceforge.net/compile.html |
| [15] Depuração no kernel: user-mode-linux.sourceforge.net/debugging.html |
| [16] Depurando o UML: user-mode-linux.sourceforge.net/debug-session.html |

Rodando Mac OS X no Linux

Uma maçã por dia...



Os emuladores agora oferecem aos usuários a capacidade de rodar o amado Mac OS X no Linux, matando vários coelhos com uma só cajadada (nenhum animal foi ferido durante a produção deste artigo).

POR OLIVER FROMMEL

O modo mais fácil de rodar programas do Mac no Linux é o software livre *Mac-on-Linux* [1]. Ele não oferece emulação de hardware e, por isso, exige um computador Apple. O pacote *Mac-on-Linux* teoricamente permite o uso de outras arquiteturas PowerPC [2], como [3], mas a instalação do sistema operacional da Apple, o Mac OS, nessas arquiteturas iria transgredir o contrato de licença.

Linux para Apple

É possível usar o *Mac-on-Linux* em computadores Apple como os iMac, iBook, Power Mac ou PowerBook. Claro, é preciso que sua distribuição Linux tenha uma versão para PowerPC [4]. As seguintes distribuições têm essa capacidade: Fedora Core 3 (ainda em fase de testes [5]), Gentoo [6], Debian [7], Ubuntu [8] e Yellow Dog [9].

O *Mac-on-Linux* não emula hardware; ele passa comandos diretamente à CPU. Também traz alguns drivers de dispositivos que foram otimizados para essa aplicação de modo semelhante ao VMware, o que explica porque o *Mac-on-Linux* é muito mais rápido que um emulador.

Mac OS como um programa do Linux

O *Mac-on-Linux* roda um sistema operacional Apple padrão como um programa do Linux. Em nosso exemplo, usaremos o Mac OS X, mas o *Mac-on-Linux* também permite o uso de variantes mais antigas, como o Mac OS 9, que ainda é bastante comum.

Usuários do Gentoo podem baixar e compilar o *Mac-on-Linux* do típico modo Gentoo usando o comando `emerge mo1`. O Yellow Dog Linux, na verdade, já inclui o programa em sua seleção padrão.

Usamos a distribuição Ubuntu para o processador PowerPC, já que vimos ser impossível

fazer com que o Gentoo ou a distribuição comercial Yellow Dog iniciassem o servidor gráfico em nosso iBook G4/800, cuja placa de vídeo usa um chipset Radeon Mobility 9200.

O instalador do Ubuntu funciona com o gerenciador de pacotes *apt* do Debian, que baixa arquivos da Internet. O *Mac-on-Linux* precisa de alguns módulos do kernel para o PPC Linux; você mesmo terá de compilar esses módulos. Isso significa instalar algumas ferramentas, se é que elas já não estão lá. O seguinte comando baixa os pacotes e os instala:

```
# sudo apt-get install build-essential \
linux-headers-2.6-powerpc
```

Se lhe pedirem uma senha, digite seu nome de usuário e senha normais, já que tipicamente o Ubuntu não tem uma conta de administrador (root, ver [8]). Ao invés disso, os usuários podem rodar comandos que tipicamente necessitam de privilégios de root usando `sudo nome_do_comando`. Obviamente você também pode instalar os

Quadro 1: Adições a `/etc/apt/sources.list`

```
deb http://archive.ubuntu.com/ubuntu warty multiverse
deb-src http://archive.ubuntu.com/ubuntu warty multiverse
```

pacotes usando o gerenciador gráfico de pacotes *Synaptic* e fazer tudo com o mouse, sem precisar ficar digitando comandinhos.

Infelizmente, o repositório padrão do Ubuntu não inclui os arquivos de que precisamos, de forma que será preciso antes de tudo adicionar à lista de fontes do APT o repositório *multiverse*, que inclui pacotes que não fazem parte da distribuição padrão do Ubuntu. Para isso, acrescente as linhas do **quadro 1** a seu arquivo `/etc/apt/sources.list`.

Agora atualize o repositório local para que aponte para o novo software, digitando `sudo apt-get update`. Em seguida, instale o código fonte para os módulos do kernel do Mac-on-Linux:

```
# sudo apt-get install mol-modules-source
```

Abra um editor de textos para editar o cabeçalho desse pacote e remover um erro que de outra forma impediria você de compilar os módulos. Se você preferir usar outro editor que não o *vi*, substitua *vi* pelo nome de seu editor favorito.

```
# sudo vi /usr/src/linux-headers-2.6.8.1-2
4-powerpc/include/asm/setup.h
```

Apague a linha com `#include <asm-m68k/setup.h>` e salve o arquivo novamente. Mude então para o diretório com o código fonte do Linux e descompacte o arquivo que o *apt-get* armazenou ali.

```
# cd /usr/src
# sudo tar xzvf mol-modules.tar.gz
```

Você precisa ajustar algumas variáveis de ambiente para se certificar de que os módulos combinam com sua versão do kernel atual:

```
# export KVERS="$(uname -r)"
# export KSRC="/usr/src/linux-2
headers-$(uname -r)"
# export KDREV="ubuntu0"
```

Agora vá ao subdiretório com o Mac-on-Linux e chame o script de compilação:

```
# cd modules/mol
# sudo debian/rules build
```

Depois de um curto intervalo, a compilação estará completa; agora você pode criar um pacote Debian com os arquivos criados por essa etapa:

```
# sudo debian/rules binary-mol-modules
```

O produto final deverá estar em `/usr/src`. Mude para esse diretório e instale o software em seu sistema Ubuntu:

```
# sudo dpkg -i mol-modules-2.6.8.1-3-2
powerpc_0.9.70+ubuntu0_powerpc.deb
```

Isso finalmente completa o trabalho de preparação e você poderá então instalar os pacotes restantes do Mac-on-Linux:

```
# sudo apt-get install mol mol-drivers-macosx
```

O gerenciador de pacotes resolve dependências por si só e adicionalmente instala o *mol*, o *mol-drivers-macosx* e o *mol-drivers-linux*, necessário apenas se você quiser rodar uma versão PPC do Linux.

Se você especificar o parâmetro `--loadonly` ao iniciar o programa, ele informa se o módulo do kernel foi carregado (**figura 1**). O engano mais comum é ter versões do kernel e do módulo que não combinam. Embora o parâmetro `-a` permita diferentes números de versão, você não deve esperar que uma configuração onde os números não combinam funcione.

```
sliver@mother:/usr/src/modules/mol $ sudo startmol --loadonly
Mac-on-Linux 0.9.70 [Aug 3 2004 16:18]
Copyright (C) 1997-2004 Samuel Hydr
Starting MOL version 0
Loading Mac-on-Linux kernel module:
  /lib/modules/2.6.8.1-3-powerpc/misc/mol.ko
sliver@mother:/usr/src/modules/mol $
```

Figura 1: Iniciando o Mac-on-Linux com a opção `--loadonly` inclui nos registros do sistema (*logs*) os módulos necessários para rodar o programa.

Depois de carregar o módulo, você pode passar à configuração do vídeo. O comando `sudo molvconfig` inicia o programa requerido, que testa sozinho os modos de operação ou permite escolher um modo manualmente. Em nosso laboratório, ambas as opções funcionaram bem com a configuração padrão.

`startmol` ajuda a organizar as atribuições do teclado. O parâmetro `--keyconfig` inicia um programa interativo, que exibe o nome da tecla (por exemplo, *Return* ou *Apple-Key*) e pede que você a pressione.

Depois de completar a configuração, você pode iniciar o Mac OS X numa janela digitando `sudo startmol -X`. O Mac OS X é imediatamente localizado no disco rígido e o sistema “inicia”. O Mac-on-Linux usa uma janela de terminal para exibir mensagens que surjam durante o boot. Essas mensagens são armazenadas permanentemente num arquivo sob `/var/log`, que é onde você deve olhar se algo der errado.

Túnel de rede

Configurar a rede é uma tarefa mais difícil e, infelizmente, não há um assistente para nos mostrar o caminho certo. Comece habilitando a interface de túnel no arquivo `/etc/mol/molrc.net`; essa interface carrega automaticamente o módulo do kernel e atribui um endereço IP. Para habilitar a interface de túnel, remova o `#` na linha

Quadro 2: `/etc/mol/tunconfig`

```
#!/bin/bash
/sbin/ifconfig tun0 192.168.40.1
/sbin/iptables -D POSTROUTING -t nat -s 192.168.40.0/24 -d ! 192.168.40.0/24 -j MASQUERADE
/sbin/iptables -t nat -s 192.168.40.0/24 -d ! 192.168.40.0/24 -A POSTROUTING -j MASQUERADE
/sbin/iptables -t filter -P FORWARD ACCEPT
echo 1 > /proc/sys/net/ipv4/ip_forward
```

com `netdev: tun0 -tun`. Agora suas mensagens de boot deverão incluir uma linha semelhante a `Ethernet Interface 'tun-<tun0>' @ 00:00:0D:EA:DB:EE`.

Um novo disco aparece no Mac OS X; abra-o com um duplo clique. Esse disco contém os drivers especiais de que o Mac OS X precisa para acessar a rede em modo de emulação. Para instalar os drivers, clique no pacote `Pkg` e siga as instruções.

Você provavelmente vai querer usar o Mac-on-Linux para acessar tanto a rede quanto a Internet. Para passar tráfego ao Mac-on-Linux, é preciso editar o script `/etc/mol/tunconfig`, que é iniciado por `startmol`. A versão padrão deste script inicia um servidor DHCP e usa o `iptables` para mapear endereços IP.

Em nosso laboratório, usamos um `tunconfig` alternativo com um endereço IP estático (ver **quadro 2**). No Mac OS X, atribua o endereço IP `192.168.40.2` à interface de rede `en3` e configure `192.168.40.1` como roteador.

Depois de terminar essa configuração, o Mac-on-Linux terá acesso à Internet.

Acelerador

Para iniciar o Mac-on-Linux em modo tela cheia no X, é preciso modificar um arquivo chamado `/etc/mol/molrc.video`. Na linha `enable_xvideo: ...`, mude o valor de `yes` para `no`. Esse valor faz com que o Mac OS X rode muito mais rápido, já que elimina a necessidade de uma porção de operações gráficas comedoras de tempo. De acordo com a documentação do Yellow Dog, a velocidade é de “quase 100%” em comparação com um sistema que roda diretamente no hardware. É possível conseguir o mesmo efeito desprezando o X e iniciando o Mac-on-Linux diretamente num console texto (ou seja, não numa janela de terminal do ambiente gráfico). Para maiores informações consulte **[10]**.

Uma desvantagem do Mac-on-Linux é ser impossível trocar os CD ROMs após iniciar o emulador. Resta a esperança de que os desenvolvedores do Mac-on-

Linux removam esse problema numa versão futura. Quem sabe você mesmo não tenha as habilidades necessárias e queira dar uma mão?

Mac OS sem Hardware Apple

Se você não possui um computador da Apple, mas ainda estiver a fim de dar uma olhada nesse colorido sistema operacional, precisa sair à cata de uma alternativa. Ou seja, você precisa de um emulador de verdade que tapeie o Mac OS X para que ele acredite estar rodando numa plataforma Apple. Também ajuda se você gostar de experimentar – e, é claro, você vai precisar de um conjunto de CDs de instalação do Mac OS X. Lembre-se de que o Mac OS X é um software comercial, portanto nada de pirataria, hein?

O *PearPC* **[11]** implementa uma CPU PowerPC e um conjunto de periféricos. A instalação é “sopa no mel”:

```
# tar xjf pearpc-0.3.1.tar.bz2
# cd pearpc-0.3.1
# ./configure --enable-ui=SDL --enable-cpu=
jitc_x86
# make
# su -c 'make install'
```

Neste exemplo, o PearPC está usando a biblioteca SDL para desenhar a área de trabalho, embora seja possível usar `--enable-ui=x11` como alternativa.

Ao término da compilação, crie um arquivo vazio, que o PearPC usará como disco virtual. O seguinte comando cria um arquivo de 3 GBytes chamado `macosx.img`:

```
# dd if=/dev/zero of=macosx.img bs=516096
seek=6241 count=0
```

Ao contrário do que dizem as instruções na documentação do PearPC, é possível particionar o disco rígido virtual sem ter de recorrer ao *Darwin OS*, que é a versão livre do sistema operacional da Apple. Porém não se anime: o Darwin inclui



Figura 2: A tela de início do PearPC permite escolher a partição de boot.

apenas os componentes em modo texto. A interface gráfica, a área de trabalho e outros componentes principais (como o Quicktime) estão faltando.

É uma boa idéia criar um novo diretório para abrigar o emulador e chamá-lo `ppc`, por exemplo. Então copie os arquivos-modelo de configuração para esse diretório com o comando:

```
# cp pearpc-0.3.1/ppccfg.example ppc/ppcfg
```

Para imitar o adaptador gráfico, o PearPC também necessita de um arquivo `video.x`, que você precisa copiar novamente para seu diretório de trabalho. Mova também a imagem do disco rígido, em nosso exemplo chamada `macosx.img`, para o mesmo diretório; assim, todos os arquivos de que você necessita estarão juntos no lugar certo.

É possível editar o arquivo de configuração `ppccfg` com um editor de texto. Os ajustes padrão funcionarão bem, mas você precisa modificar o nome da imagem de seu disco rígido. Para isso, mude uma linha no arquivo, como segue:

```
pci_ide0_master_image = "macosx.img"
```

A linha `prom_bootmethod` especifica como o PearPC inicia. O valor pré-configurado, `auto`, usa a primeira partição de boot. Se você mudar esse valor para `select`, o emulador esperará que você selecione uma partição (**figura 2**).

Insira o primeiro CD de instalação do Mac OS X para começar a instalação. Se você já não estiver no diretório `ppc`, vá até ele e inicie o emulador, especificando seu arquivo de configuração como parâmetro:

```
# ppc ppcfg
```

Se você escolher iniciar a partir do CD, a tela de início do Mac OS X aparecerá, como mostrado na **figura 3**. Você pode pressionar **[Alt]+[Enter]** para alternar entre os modos de exibição em tela cheia e janela.

Em alguns instantes, aparece o programa de instalação. Se você atravessar as etapas necessárias, vai precisar definir um “volume alvo” em determinado momento, mas infelizmente você ainda não tem um. É preciso antes particionar o disco rígido virtual. Para iniciar a ferramenta de particionamento, selecione o *Installer | Hard disk tool*, no menu no topo da tela.

Na ferramenta de disco rígido clique na aba *Partitioning* e escolha a opção *1 Partition* no item *Volume Schema*. Clique em *Partition* e confirme a operação na tela seguinte. O programa particiona o disco virtual e formata a nova partição. Após sair da ferramenta, a partição estará visível no instalador.

Se o instalador se queixar de pouco espaço em disco, é possível desselecionar alguns pacotes, como os *Language packages* e *Additional voices*. Ao fim da seleção de pacotes, o instalador os copiará para



Figura 3: O instalador do Mac OS X inicia no PearPC como se você estivesse num Mac de verdade.

seu disco virtual. No final dessa etapa, saia do PearPC e insira o segundo CD de instalação. Ao reiniciar, selecione a partição criada anteriormente como partição de boot. O instalador copiará mais alguns arquivos e completará a instalação. Você pode pular as questões sobre o registro de sua conta de usuário.

Redes para especialistas

Para permitir que o Mac OS X use a rede num ambiente PearPC, é preciso modificar a linha referente à placa de rede no arquivo de configuração:

```
pci_rt18139_installed = 1
```

Ao reiniciar, o emulador deve reclamar de que está faltando uma interface de túnel de rede (*network tunnel interface*) `/dev/net/tun`. Se isso acontecer, rode `modprobe tun` como *root* para carregar o módulo do kernel do Linux.

A placa Ethernet virtual é automaticamente detectada pelo Mac OS X; verifique a configuração de rede nas Preferências do Sistema (*Maçã | Preferências do Sistema*). Atribua o endereço IP estático **192.168.1.1**, que foi configurado em `scripts/ifppc_up`. O endereço do roteador é **192.168.1.80**. Especifique um servidor DNS funcional no arquivo `/etc/resolv.conf`.

Assim como no Mac-on-Linux, o PearPC configurará a rede virtual de maneira bastante autônoma em segundo plano e até mesmo ajustará a tradução de endereços por NAT (*Network Address Translation*). Para isso, basta copiar todo o subdiretório `scripts` para o lugar certo:

```
# cp -R pearpc-0.3.1/scripts ppc
```

Como os programas mudam as configurações de rede do Linux, você precisa de privilégios de administrador para essa etapa. Em outras palavras, você precisa iniciar o PearPC como administrador ou atribuir privilégios SUID de root para os

programas `ifppc_up.setuid` e `ifppc_down.setuid` (use o comando `chmod +s nome_do_programa`).

Por padrão, o PearPC cria uma interface de rede chamada `ppc` e um endereço **192.168.1.80** no sistema Linux; chame `/sbin/ifconfig ppc` para verificar esse ponto.

Uma união feliz

O Mac-on-Linux e o PearPC impressionam por causa de sua riqueza de recursos e facilidade de uso. O fato de o Mac-on-Linux superar de longe seu competidor não é nem um pouco surpreendente. Afinal, estritamente falando, o Mac-on-Linux não é um emulador, mas um programa que roda o Mac OS X no Linux. De fato, ele é rápido o bastante para ser usado em trabalhos sérios num computador moderno. O modo como o programa lida com CDs é um problema. Nem o Mac-on-Linux nem o PearPC permitem que os usuários troquem os CDs depois de iniciados.

Ambos os programas têm uma configuração de rede bastante complicada, mas isso se relaciona mais à complexidade das funções de que o Linux precisa para permitir a conexão de rede virtual. Entretanto, isso deve ser corrigido no futuro. ■

INFORMAÇÕES

- | | |
|------|---|
| [1] | Mac-on-Linux: www.maconlinux.org |
| [2] | Processador PowerPC: www.ibm.com/chips/products/powerpc |
| [3] | Pegasos Computer www.pegasosppc.com |
| [4] | Site oficial do Linux no PowerPC: www.penguinppc.org |
| [5] | Fedora Core 3 PPC: fedoraproject.org/fedorappc/FC-3 |
| [6] | Gentoo PPC: www.gentoo.org/doc/en/handbook/handbook-ppc.xml |
| [7] | Debian PPC: www.debian.org/ports/powerpc |
| [8] | Ubuntu: www.ubuntulinux.org |
| [9] | Yellow Dog: www.yellowdoglinux.com |
| [10] | Modo tela cheia no Mac-on-Linux: www.maconlinux.org/userguide/cvideo.html |

Criando máquinas virtuais de alta performance.

Multiplicando Pingüins

O conceito de distribuir os recursos de um único computador em diversas instâncias de software não é recente. A origem da virtualização de computadores vem da época dos mainframes, computadores de grande porte destinados a rodar várias instâncias de um sistema operacional, cada qual atendendo a um usuário ou finalidade específica.



POR MARCO ANTONIO BOCARDO SINHORELI

Diferentemente de outros softwares destinados a dividir seu hardware, o Xen consegue, de maneira muito tranqüila, dar a ilusão de que existem diversas máquinas em sua rede, com praticamente o mesmo desempenho de uma máquina Linux real. Estudos comparativos demonstram que o Xen, em diversas situações, é superior em desempenho a softwares comerciais e Open Source, como você pode comprovar no site oficial do programa [1].

O Xen é um monitor de máquina virtual para hardware x86, desenvolvido e mantido pela Universidade de Cambridge, capaz de suportar a execução de diversos sistemas operacionais sem precarizar ou prejudicar o desempenho e com isolamento total entre as máquinas virtuais, garantindo assim maior segurança. O Xen, atualmente na versão 2.0.5 (estável), tem suporte às versões 2.4.x e 2.6.x do kernel Linux.

A comunidade tem contribuído bastante no que diz respeito à criação de versões para outros sistemas operacionais. Atualmente existem versões para o NetBSD, FreeBSD e Plan 9 [2]. A versão para Windows XP está sendo baseada numa versão instável (ainda em desen-

volvimento) do Xen e não está disponível por causa de duas restrições de licenciamento do software.

Recursos do Xen

O Xen necessita de um “pré-kernel” no Linux hospedeiro para disponibilizar as máquinas virtuais. No momento do boot, o GRUB descompacta esse pré-kernel e define os parâmetros relativos à fatia de memória que o kernel linux da máquina hospedeira irá gerenciar. Esse espaço reservado para o kernel privilegiado destina-se ao uso pelos serviços gerenciados do Xen e outros inicializados pelo *init*. Cada máquina virtual (ou *domínio*) tem seu próprio arquivo de configuração (sob `/etc/xen/auto` – falaremos dele mais adiante), podendo haver variações no hardware disponível (processador, memória, swap, e discos).

O Xen utiliza por padrão o *scheduler* BVT. Para que você entenda melhor, o *scheduler* é usado para atribuir o peso correto para cada máquina virtual no processador central, conforme o peso atribuído para o domínio. O processador central nunca estará inativo se houver tarefas a serem executadas por um domínio. O BVT faz uso do “tempo virtual” para

tomar decisões de como deverá ocorrer a distribuição dos processos dos domínios no processador. Cada domínio tem um tempo e uma fatia do processador associada em nano-segundos.

O *Atropos Scheduler* (`sched=atropos`, no arquivo de configuração) é um *scheduler* de tempo real leve. Ele garante o compartilhamento absoluto da CPU, com a vantagem de compartilhar o tempo ocioso do processador central quando ele não está em uso. Ao usar o *Atropos*, não o configure para utilizar 100% ou mais do processador. Tente utilizar sempre uma quantidade menor que a capacidade existente para garantir um comportamento regular.

O *Round Robin Scheduler* (opção `sched=rrobin`) está incluído como uma demonstração simples de uma API interna de *scheduler* do Xen e não é indicado para uso em produção.

Disponibilizando VBDs (Virtual Block Devices) para as máquinas virtuais

É possível exportar diretamente de *Dom0* (a máquina hospedeira) dispositivos de blocos para os outros domínios, utilizando protocolos de rede padrão (NBD, iSCSI,

NFS etc), além de volumes e partições físicas para prover o diretório raiz para as máquinas virtuais.

Exportar diretamente partições de *Dom0* para os outros domínios é simples: os arquivos de configuração dos domínios utilizam a *tag phy*: para especificar um dispositivo físico a ser exportado. Exemplo da linha a ser utilizada:

```
disk = ['phy:hda3,sda1,w']
```

Com isso, dizemos que */dev/hda3* será o volume físico a ser exportado para o domínio em modo de leitura e escrita como */dev/sda1*.

Qualquer disco ou partição considerado como um dispositivo de bloco pelo Linux poderá ser exportado para os domínios virtuais. Por exemplo, se temos discos iSCSI ou volumes NBD montados em *Dom0*, eles poderão ser exportados para os domínios utilizando a sintaxe *phy*:. Veja um exemplo:

```
disk = ['phy:vg/lvm1:sda2:w']
```

Utilizando "loop devices" como VBD

É possível criar em *Dom0* os chamados *loop disks* arquivos que são montados e tratados como discos rígidos reais para serem usados como dispositivo de armazenamento das máquinas virtuais. O primeiro passo é criar os tais *loop disks*. Por exemplo, para criar um disco de 1 GB digite o comando:

```
# dd if=/dev/zero of=/loopdisk bs=1k \
seek=1024k count=1
```

Após isso será necessário formatá-lo:

```
# mkfs -t ext3 loopdisk
```

Caso o comando solicite confirmação, responda **Y**. O arquivo de configuração das máquinas virtuais ficará assim:

```
['file:/caminho/para_o/arquivo/loopdisk,\
sda1,w']
```

Existem limitações que poderão influenciar na decisão de qual tipo de armazenamento usar. Um deles é a alta requisição I/O por parte das máquinas virtuais no dispositivo físico de *Dom0* onde está o *loop disk*. Outro detalhe: por padrão, o kernel Linux define que o número máximo de *loop devices* que podem ser montados simultaneamente é oito. Esse limite é definido pelo parâmetro *max_loop* do módulo *loop.o*. Caso desejemos mais VBDs, devemos providenciar algumas alterações. O meio mais fácil é compilar o módulo *loop.o* diretamente no kernel (opção *CONFIG_BLK_DEV_LOOP* na configuração do kernel) e passar o parâmetro *max_loop=n* (onde *n* é o número máximo de *loop devices* montados) para o kernel pelo *bootloader*.

Utilizando volumes LVM como VBDs

Um volume LVM (*Logical Volume Manager*) é uma solução bastante agradável para uso como VBD, principalmente por possibilitar o redimensionamento dinâmico do volume, permitindo o ajuste do espaço em disco disponível de acordo com a necessidade. Para inicializar uma partição com suporte a um PV (*Physical Volume* – Volume físico) LVM digite:

```
# pvcreate /dev/sda6
```

Com o comando abaixo, crie um VG (*Volume Group*) com um nome qualquer, por exemplo *vg*, na partição física:

```
# vgcreate vg /dev/sda6
```

Se desejar inserir um outro disco ou partição no grupo digite:

```
# pvcreate /dev/sdb1
# vgcreate vg /dev/sdb1
```

Assim as duas partições pertencerão ao mesmo VG e serão vistas como um único dispositivo.

Agora crie um LV (*Logical Volume* – Volume lógico) com o tamanho desejado, neste exemplo 4 GB:

```
# lvcreate -L4096M -n vmdisk1 vg
```

Vamos formatá-lo com o comando:

```
# mkfs -t ext3 /dev/vg/vmdisk1
```

Agora basta indicar o VBD no arquivo de configuração do domínio virtual:

```
disk = [ 'phy:vg/vmdisk1,sda1,w' ]
```

Você pode clonar volumes LVM. Isto é possível desde o kernel 2.6.8; no entanto, esta versão possui um bug, já corrigido nas mais recentes. Para utilizar esse recurso basta digitar:

```
# lvcreate -s -L1024M -n clonedisk1 \
/dev/vg/myvmdisk1
# lvcreate -s -L1024M -n clonedisk2 \
/dev/vg/myvmdisk1
```

Assim criamos dois clones de 1 GB do volume */dev/vg/myvmdisk1*. Se tiver de redimensioná-los use o utilitário *lvextend*:

```
# lvextend +100M /dev/vg/clonedisk1
```

Nesse caso acrescentamos mais 100 MB ao volume */dev/vg/clonedisk1*.

Usando compartilhamentos NFS como VBDs

Inicialmente será necessário um sistema de arquivos exportado em um servidor NFS. Isso pode ser feito adicionando a linha a seguir ao arquivo */etc/exports*:

```
/exports/rootvm1 1.2.3.4/24 (rw, sync, \
no_root_squash)
```

Finalmente, no arquivo de configuração da máquina virtual (veja [listagem 2](#)), indique que deseja usar um compartilhamento NFS como partição de root :

```
root = '/dev/nfs'
nfs_server = '1.1.1.1'
nfs_root = '/exports/rootvm1'
```

A máquina virtual precisa de acesso à rede no momento do boot. Para isso você deve atribuir a ela um endereço IP, seja manualmente (com os utilitários `ip`, `netmask`, `gateway` e `hostname`) ou automaticamente via DHCP, com o parâmetro `dhcp = 'dhcp'` no arquivo de configuração da máquina virtual.

É importante lembrar que um sistema de arquivos raiz montado via NFS tem problemas de estabilidade sob uma alta carga de requisições (e esse problema não é específico do Xen). Nesse caso, em aplicações de missão crítica o uso de NFS não é indicado.

Um cenário de uso

Imagine que você precisa disponibilizar diversos serviços em dez servidores diferentes sendo que cada servidor teria, conforme o serviço que será fornecido, uma certa quantidade de memória e espaço em disco. Considere também que na maior parte do tempo os processadores destes servidores estarão ociosos. Além disso seus servidores precisarão estar ligados à rede, mas precisam também ter um certo nível de segurança.

Tradicionalmente esse cenário exigiria grande investimento em infraestrutura de servidores e outros ativos de rede, o que poderia inviabilizar a execução de seu projeto. Mas se em vez de várias máquinas tivermos um único servidor com mais poder de processamento e bastante memória e espaço em disco, o Xen poderá tranquilamente satisfazer esse cenário com maior qualidade, mais segurança e muita economia de recursos. O

investimento em infraestrutura sofreria uma redução drástica, possibilitando a implantação da solução.

O cenário acima descrito necessita dos seguintes itens:

- servidores Apache com 5 GB de espaço em disco e 256MB de RAM.
- servidor MySQL com 3 GB de espaço em disco e 256MB de RAM.
- servidor PostgreSQL com 2 GB de espaço em disco e 512MB de RAM.
- servidor SMTP com 3 GB de espaço em disco e 512MB de RAM.
- servidor POP3/IMAP com 10 GB de espaço em disco e 512MB de RAM.
- servidor LDAP com 2 GB de espaço em disco e 256 de RAM.

Teremos que dimensionar o Xen para que cada máquina virtual forneça a quantidade adequada de espaço em disco e memória. Teremos ainda que providenciar para que o sistema Linux hospedeiro, que gerenciará as máquinas virtuais, tenha quantidade de memória e espaço em disco suficientes para a tarefa. Precisaremos da seguinte configuração: para a memória das máquinas virtuais, (256 MB x 7) + (512 MB X 3), ou 3328 MB de RAM, e (5 GB x 5) + (2 GB x 2) + (3 GB x 2) + 10 GB = 45 GB de espaço em disco. Além disso, precisaremos de 512 MB de RAM e 5 GB de espaço em disco para o sistema hospedeiro. Ou seja, precisaremos de uma máquina com 3.75 GB de RAM e 50 GB de espaço em disco para implementar nosso cenário.

Como teremos concorrência entre as várias instâncias no que diz respeito ao uso do processador, a melhor alternativa é disponibilizar um servidor capaz de suportar a alta latência de requisições sem degradação dos serviços. Para o cenário anteriomente descrito, utilizaremos o scheduler padrão do Xen, o BTV, para que o próprio virtualizador faça dinamicamente a distribuição dos processos em execução em todas as máquinas virtuais.

Se seus requisitos de hardware forem menores que os apresentados, será possível implementar tranquilamente uma solução com hardware menos potente. Os processadores suportados pelo Xen são os da família i686 ou mais recentes (Intel Pentium Pro, Celeron, Pentium II, Pentium III, Pentium IV, Xeon, e AMD Athlon ou Duron). Máquinas com dois ou mais processadores são suportadas e há suporte básico à tecnologia *Hyper-Threading* (SMT) da Intel. Atualmente o Xen opera somente em modo 32 bits, mas uma versão específica para a arquitetura x86/64 (usada nos processadores Opteron, da AMD) está sendo providenciada e uma para a arquitetura Itanium está em vias de conclusão (ambas são arquiteturas de 64 Bits). Estuda-se ainda adicionar suporte a outras arquiteturas, como PowerPC e ARM, no momento adequado.

Preparando-se para a multiplicação

O primeiro passo para a multiplicação dos pingüins é fazer o download dos fontes da versão estável mais recente do Xen (em [\[3\]](#)). No momento em que este artigo foi escrito, o Xen encontrava-se na versão 2.0.5. Precisaremos ainda resolver algumas dependências antes de iniciar a compilação do programa.

Para escrever este artigo, usei o Debian GNU/Linux. Encontrei no site do projeto, na sessão FAQ [\[4\]](#), a lista de tudo o que precisaria instalar antes de dar início à compilação do Xen. Os pacotes no Debian são: `make`, `gcc`, `libc6-dev`, `zlib1g-dev`, `python`, `python-dev`, `python-twisted`, `bridge-utils`, `iproute`, `libcurl3`, `libcurl3-dev`, `bzip2`, `module-init-tools`, `latex`, `latex2html`, `transfig` e `tgif`. O comando é:

```
# aptitude install make gcc libc6-dev 2
zlib1g-dev python python-dev iproute 2
python-twisted bridge-utils libcurl3 2
libcurl3-dev bzip2 module-init-tools 2
latex latex2html transfig tgif
```

O script de instalação providenciará os fontes do kernel, que você deverá indicar no `Makefile`. Nesse arquivo, considere que `xen0` se refere ao kernel do sistema Linux hospedeiro e `xenU` é referente ao kernel da máquina virtual. O `U` após `xen` significa *Unprivileged Kernel* (kernel sem privilégios). Exemplo:

```
# Para kernels da série 2.6.x
KernelS ?= linux-2.6-xen0 linux-2.6-xenU
# Para kernels da série 2.4.x
KernelS ?= linux-2.4-xen0 linux-2.4-xenU
```

Podemos ainda fazer variações colocando uma versão do kernel em `xen0` e outra em `xenU`, de acordo com suas necessidades. Rode o comando `make` para compilar o programa:

```
# make install-twisted all install
```

Isso vai colocar em `/boot` os kernels do hospedeiro e das máquinas virtuais. Podemos também personalizar o kernel do Xen para satisfazer alguma necessidade específica. Para o kernel das máquinas virtuais digite:

```
# cd linux-2.6.10-xenU
# make ARCH=xen menuconfig
```

Para o kernel hospedeiro digite:

```
# cd linux-2.6.10-xenU
# make ARCH=xen menuconfig
```

Configurando o gerenciador de boot

Infelizmente, para os fãs do LILO, o GRUB é, até o momento, o único gerenciador de boot que suporta as opções do Xen usadas para determinar a quantidade de memória para o sistema hospedeiro, entre outras coisas. A entrada de nosso hospedeiro no arquivo de configuração do GRUB (que fica em `/boot/grub/menu.lst`) fica assim:

```
title Xen 2.0 / XenLinux 2.6.10
  root      (hd0,1)
  kernel /boot/xen.gz dom0_mem=524288
  module /boot/vmlinuz-2.6.10-xen0
  root=/dev/hda2 ro console=tty0
```

Vamos entender as coisas. A linha:

```
kernel /boot/xen.gz dom0_mem=524288
```

Diz que `/boot/xen.gz` é o kernel de baixo nível do Xen, o "pré-kernel" mencionado anteriormente. Ele é utilizado para levantar o kernel privilegiado de `dom0` (o hospedeiro), que é o monitor Xen. A opção `dom0_mem=524288` define a quantidade de memória que o hospedeiro Xen irá utilizar, neste caso 512 MB. Já a linha:

```
module /boot/vmlinuz-2.6.10-xen0 root=
/dev/hda2 ro console=tty0
```

O parâmetro `module /boot/vmlinuz-2.6.10-xen0` define o kernel que o monitor (sistema Linux hospedeiro) irá utilizar. `root=/dev/hda2 ro` define a partição raiz de `dom0`, montada inicialmente no modo `ro` (*read-only*, somente para leitura). O parâmetro `console=tty0` especifica o terminal de I/O (entrada/saída) padrão do Xen. Veja mais opções em [6].

Para usar o Xen com kernels da série 2.6.x, é necessário desabilitar o suporte a TLS (*Thread Local Storage*) antes de carregar o kernel hospedeiro. Para isso mova o diretório `/lib/tls` para `/lib/tls.disabled`. A equipe de desenvolvimento do Xen vem trabalhando juntamente com os mantenedores das distribuições Linux para que essa incompatibilidade com as bibliotecas TLS seja resolvido.

Criando as máquinas virtuais

Para criar o sistema de arquivos raiz para nossas máquinas virtuais, usaremos o utilitário `debootstrap`. Instale-o em sua máquina (`apt-get install debootstrap`) e digite o seguinte:

```
# mkdir /usr/src/rootfs
# debootstrap --arch i386 sarge
/usr/src/rootfs http://ftp.br.debian.org/debian
```

Teremos, então, um esqueleto de um sistema básico Debian em `/usr/src/rootfs`. Será agora necessário copiar os módulos compilados do kernel `xenU` para o novo sistema de arquivos:

```
# cp -ax /lib/modules/2.6.10-xenU
/usr/src/rootfs/lib/modules
```

Conforme dissemos anteriormente, a solução com mais flexibilidade para criar os VBDs, ou discos, das máquinas virtuais é o LVM. Caso o hospedeiro use um kernel da série 2.4 use o pacote `lvm10`, mas você não terá acesso ao recurso de clonagem de discos. Caso use um kernel da série 2.6 opte pelo `lvm2`. Em nosso exemplo, `/dev/sda`, que tem 40 GB, conterá as partições do hospedeiro e parte do volume LVM. O restante ficará em um volume idêntico, chamado `/dev/sdb`. Ative os dispositivos físicos que vamos usar:

```
# pvcreate /dev/sda4
# pvcreate /dev/sdb1
# pvcreate /dev/sdb2
```

A seguir criaremos dois *Volume Groups*, um para os discos e outro para o swap:

```
# vgcreate vgdisk /dev/sda4 /dev/sdb1
# vgcreate vgsnap /dev/sdb2
```

Agora chegou a hora de criar os volumes lógicos. Fizemos um script (veja a [listagem 1](#)) para automatizar a tarefa de criação dos volumes para os cinco servidores Apache.

Agora criaremos os volumes lógicos para o servidor MySQL, que tem 3 GB de espaço em disco, e copiaremos para ele o sistema de arquivos "esqueleto" criado com o `debootstrap`:

Listagem 1: Criando volumes lógicos

```
#!/bin/bash

for ((v=1;v<=5;v++))
do
# Cria o volume lógico
lvcreate -L5120M -n apache$v vdisk
# Formata o volume em ext3
mkfs -t ext3 /dev/vdisk/apache$v
# Monta o volume em /mnt
mount -t ext3 /dev/vdisk/apache$v /mnt
# Copia os esqueleto criado com o debootstrap para o volume
cp -ax /usr/src/rootfs/* /mnt
# Desmonta o volume
umount /mnt
# Cria o volume para swap
lvcreate -L256M -n apache$v vgsdap
# Formata o volume recém-criado como área de swap
mkswap /dev/vgsdap/apache$v
done
```

```
# lvcreate -L3072M -n mysql vdisk
# lvcreate -L256M -n mysql vgsdap
# mkfs -t ext3 /dev/vdisk/mysql
# mkswap /dev/vgsdap/mysql
# mount -t ext3 /dev/vdisk/mysql /mnt
# cp -ax /usr/src/rootfs/* /mnt
# umount /mnt
```

Arquivos de configuração das máquinas virtuais

O arquivo de configuração do *daemon* do Xen, `xend-config.sxp` fica no diretório `/etc/xen/`, junto com os subdiretórios `auto` e `scripts`, além de dois arquivos de exemplo. Entre outros parâmetros, nesse arquivo é definida a porta da interface web para controle do daemon (`xend-port 8000`), os hosts que poderão se conectar a essa interface (`xend-address 'localhost'`, deixe vazio para não restringir o acesso), o modo de funcionamento das interfaces de rede virtuais (roteamento ou *bridge*), as definições de segurança para as interfaces de rede (*antispoof*) e o uso de scripts para *file-backend block device* e *enbd*.

No diretório `auto` ficam os arquivos de configuração das máquinas virtuais que serão inicializadas automaticamente no momento em que o script `/etc/init.d/xendomains` for executado. O diretório `scripts` contém os arquivos de scripts do daemon Xen.

Veja na **listagem 2** o arquivo de configuração de nossa primeira máquina virtual, que vai ser usada como servidor Web rodando o Apache. Para as demais máquinas, considere os valores declarados em nosso cenário inicial, lembrando que esses arquivos de configuração deverão estar no diretório `/etc/xen/auto`, se considerarmos que as máquinas virtuais deverão iniciar automaticamente quando o script `/etc/init.d/xendomains` for executado. O **quadro 1**, na página ao lado, resume os principais comandos que podem ser usados para operar o Xen.

Conclusão

Apesar de a instalação e configuração do Xen apresentarem um certo nível de dificuldade, o sistema não deixa nada a desejar do ponto de vista da otimização dos recursos computacionais. Conforme mencionamos neste artigo, em comparação com outros aplicativos ou sistemas de virtualização o Xen,

Listagem 2: arquivo de configuração de uma máquina virtual

```
# Define o Kernel do domínio
kernel = "/boot/vmlinuz-2.6.10-xenU"
# Define o Sistema Operacional do domínio
builder='linux'
# Quantidade de memória
memory = 256
# Nome deste domínio
name = "Apache1"
# -l determina a auto-alocação de CPU para este domínio
cpu = -1
# Número de interfaces de rede deste domínio
nics = 1
# Define o Mac Address e a bridge para este domínio. Se o Mac não for configurado, um número hexadecimal é gerado aleatoriamente.
vif = [ 'mac=aa:00:00:00:00:01, bridge=xen-br0' ]
# VBDs para este domínio.
disk = [ 'phy:vgdisk/apache1,sda1,w', 'phy:vgsdap/apache1,sda2,w' ]
# Configuração de rede. É possível fazê-la automaticamente via DHCP.
# dhcp="dhcp"
# "rootdisk", passado no momento do boot para o kernel.
root = "/dev/sda1 ro"
# Run level
extra = 2
restart = 'onreboot'
```

pelo fato de trabalhar mais próximo ao hardware, utiliza todos os seus recursos – o sistema, como vimos, lança mão de um pré-kernel para fazer chamadas de baixo nível e realizar a comunicação entre os dispositivos do sistema. Isso ocorre a partir do *boot*, quando os parâmetros do *grub* para fatar a memória para o *Xen0* (Linux hospedeiro/monitor) são passados. As funções então já alocadas na memória do pré-kernel possibilitam a criação das demais máquinas virtuais.

Para quem busca uma solução de virtualização livre e robusta, com alto desempenho e qualidade, o uso do Xen é, sem dúvida, mais que indicado. ■

INFORMAÇÕES

- [1] Comparativos de desempenho entre o Xen e outras máquinas virtuais:
www.cl.cam.ac.uk/Research/SRG/netos/xen/performance.html
- [2] Plan 9 - Um Sistema Operacional Unix mantido pela Bell Labs:
www.cs.bell-labs.com/plan9dist
- [3] Linux Magazine Brasil, 2ª. edição, Página 38, *Jogue seus dados*.
- [4] Download do Xen:
www.cl.cam.ac.uk/Research/SRG/netos/xen/downloads.html
- [5] Pacotes do Debian necessários para o Xen:
tinyurl.com/ce688
- [6] Manual do Usuário do Xen:
www.cl.cam.ac.uk/Research/SRG/netos/xen/readmes/user/user.html#s:xboot
- [7] Página oficial do Xen:
www.cl.cam.ac.uk/Research/SRG/netos/xen
- [8] The Unnofficial Xen Wiki:
xen.terrabox.com/
- [9] Linux Magazine Brasil, 7ª. Edição, Página 19, *IBM abre código de virtualização*.
- [10] Xen na Wikipedia:
en.wikipedia.org/wiki/Xen
- [11] Máquinas virtuais:
en.wikipedia.org/wiki/Virtual_machine
- [12] Comparativo entre máquinas virtuais:
en.wikipedia.org/wiki/Comparison_of_virtual_machines

Quadro 1: comandos relacionados ao Xen

Iniciar, parar, verificar o estado ou reiniciar o daemon Xen, ou forçar a re-leitura de seus arquivos de configuração do programa:

```
# /etc/init.d/xend {start|stop|status|restart|reload|force-reload}
```

Iniciar, parar ou verificar o estado dos domínios cujos arquivos de configuração podem ser encontrados no diretório `/etc/xen/auto`

```
# /etc/init.d/xendomains {start|stop|status}
```

Argumentos do xm (*Xen monitor*)

call	Chama as funções da API Xen.
Help	Mostra a ajuda do comando xm.

Comandos relacionados aos consoles das máquinas virtuais

console	Abre um console na máquina virtual. ex: <code>xm console apache1</code>
consoles	Solicita informações sobre as máquinas virtuais.

Comandos dos domínios

balloon	Configura memória nos domínios utilizando o driver <i>balloon</i> .
create	Cria um domínio.
destroy	Encerra a execução de um domínio imediatamente.
domid	Converte o nome do domínio em seu ID.
domname	Converte o ID de um domínio em seu nome.
list	Mostra informações sobre os domínios.
maxmem	Configura o limite de memória para o domínio.
migrate	Migra um domínio para outras máquinas.
pause	Pausa a execução de um domínio.
pincpu	Configura o PIN de um domínio para o uso da CPU.
restore	Reinicia a execução de um domínio a partir de seu estado salvo.
save	Salva o estado de um domínio em seu arquivo de configuração.
shutdown	Desliga um domínio.
sysrq	Envia um <i>sysrq</i> para o domínio.
unpause	Retoma a execução de um domínio pausado.

Comando relacionados ao host xen (hospedeiro)

dmesg	Lê ou limpa as mensagens no buffer do Xen.
Info	Solicita informações sobre o host Xen.
Log	Imprime o log do daemon <i>xend</i> .

Comandos de controle dos schedulers

atropos	Configura parâmetros do scheduler atropos.
bvt	Configura parâmetros do scheduler BVT.
bvt_ctxallow	Dá permissão de "context switch" ao scheduler BVT.
rrobin	Configura o scheduler <i>round robin</i> .

Comandos relacionados aos Virtual Block Devices

vbd-create	Cria um novo VBD para um domínio.
Vbd-destroy	Destrói um VBD de um domínio.
Vbd-list	Lista os VBDs de um domínio.

Comandos relacionados às interfaces de rede virtuais

vif-list	Lista as interfaces de rede virtuais.
----------	---------------------------------------

Um editor de HTML com interface integrada de desenvolvimento

Peixes na rede

O editor de HTML Bluefish é útil para projetos web em larga escala e também para projetos menores de programação. O realce da sintaxe e os blocos de código ajudam os desenvolvedores a construir projetos de forma eficiente.

POR ANDREAS GRYTZ

A construção de projetos web complexos com suporte a CGI pode ser algo muito confuso. Um editor HTML como o *Bluefish* [1], com suporte a gerenciamento de projeto e às mais populares linguagens de script ajuda você a se manter na linha. A versão atual, 1.01, oferece aos usuários alguns interessantes recursos novos.

O programa não é oficialmente parte do projeto Gnome, mas é usado com frequência em sistemas baseados neste ambiente de trabalho, pois usa o mesmo conjunto de elementos de interface (*widgets*), o *GTK+*. O *Bluefish* baseia-se na biblioteca gráfica *GTK 2.0* ou mais atual. A biblioteca *libpcre*, na versão 3.0 ou posterior, fornece o sempre útil realce da sintaxe (*syntax highlighting*); Já o corretor ortográfico, baseado no pacote *Aspell*, é opcional.

Além das dependências que já mencionamos, o software precisa de algumas bibliotecas do Gnome. Repetimos: a maioria das distribuições modernas deve ter essas bibliotecas. O programa não demanda muito espaço em disco e faz bom uso

dos recursos de memória, o que o torna muito útil para programadores que estejam trabalhando em hardware mais antigo.

Os usuários podem acessar recursos online, como servidores FTP ou diretórios WebDAV, facilmente pelo *Gnome VFS*, uma camada de abstração do sistema de arquivos. O *Bluefish* acessa os recursos como se fossem diretórios locais.

Projeto da interface

A interface do *Bluefish* tem três painéis: uma barra de menu com funções padrão no alto da janela, um painel principal e uma barra lateral abaixo dele. O painel principal é onde os usuários editam os arquivos.

A janela do *Bluefish* usa abas para permitir que os usuários alternem rapidamente entre diversos documentos. De acordo com os desenvolvedores, o programa pode lidar com até 3500 documentos simultaneamente, embora essa situação seja bastante improvável sob condições normais de produção.

A barra lateral cumpre três tarefas: primeiro, tem um pequeno navegador, que facilita o acesso dos usuários a seus

arquivos. Segundo, os desenvolvedores acrescentaram documentação para parte das linguagens de script e de web suportadas pelo programa; se necessário, os usuários podem exibir a documentação na barra lateral. Terceiro, podem-se acrescentar atalhos para diretórios frequentemente usados a essa barra.

No alto da janela do editor há um menu de abas que oferece acesso rápido aos blocos de construção do HTML. As abas são organizadas por função, permitindo aos usuários adicionar tabelas, formulários ou elementos CSS com um esforço mínimo. No caso de construções mais complexas como formulários, o programa pede ao usuário que especifique os parâmetros necessários para a exibição correta do elemento.

Abaixo das abas, o software tem algumas ferramentas adicionais para edição de scripts. Dependendo da linguagem, o menu traz blocos de construção comuns

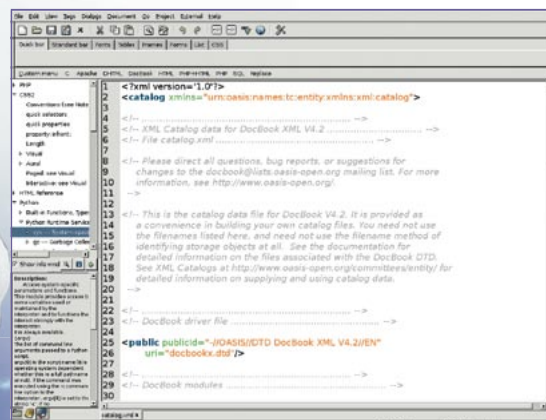


Figura 1: O *Bluefish* tem uma interface com três painéis que pode ser configurada de acordo com suas necessidades.

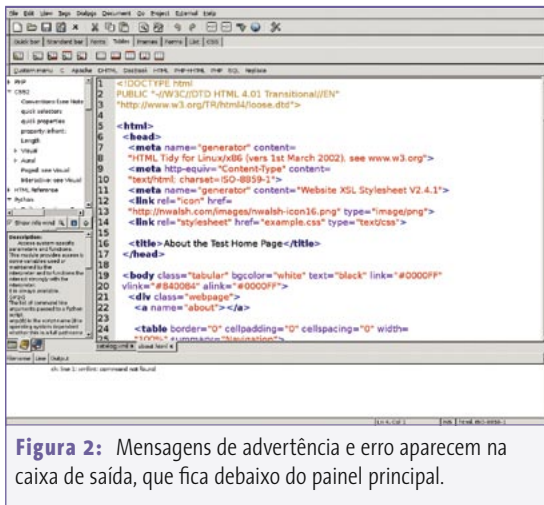


Figura 2: Mensagens de advertência e erro aparecem na caixa de saída, que fica debaixo do painel principal.

para laços, *includes* ou declarações SQL. Isso permite organizar projetos menores apontando, clicando e preenchendo caixas de diálogo.

O editor também facilita a localização/internacionalização das páginas sendo construídas. Basta clicar com o botão direito na janela do editor para alternar entre diversas variantes de codificação de caracteres. Ao salvar arquivos, o programa usa Unicode por padrão.

Um ambiente de desenvolvimento leve

Os desenvolvedores não encaram seu projeto apenas como um editor de HTML. O fato de que o Bluefish reconhece linguagens de script e linguagens tradicionais de programação como C significa que o programa se assemelha mais a um ambiente

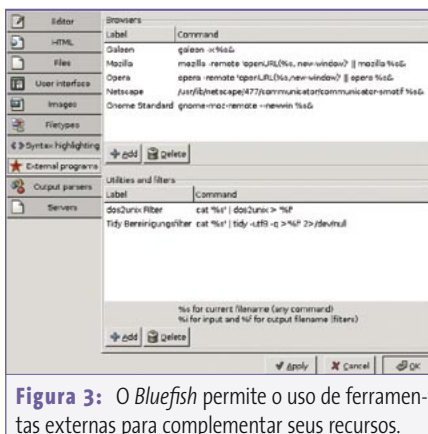


Figura 3: O Bluefish permite o uso de ferramentas externas para complementar seus recursos.

de desenvolvimento integrado (IDE – *Integrated Development Environment*) administrável do que a um simples editor HTML. Isso é útil nos casos em que um IDE com inúmeros recursos fosse atrapalhar mais do que ajudar – como seria o caso com projetos menores – já que um IDE com frequência oferece aos programadores recursos indesejados que só servem para bagunçar o coreto.

A busca fácil pela documentação integrada sobre *Python* e

PHP na barra lateral é outro recurso muito útil. Ali é o lugar certo para encontrar descrições breves das funções padrão e seus valores de retorno. Recuo automático, numeração das linhas e realce de sintaxe sofisticado facilitam o trabalho com o código fonte.

O Bluefish permite aos usuários avaliar *makefiles* e exibir as chamadas numa janela à parte. Usa o mesmo método para chamadas a programas externos, como o *Tidy*, um inspetor de sintaxe HTML [2].

Alguns programas externos são pré-configurados pela instalação padrão do Bluefish; porém, a rotina de instalação não confere se os programas estão realmente instalados e nos caminhos especificados. Isso pode causar confusão, já que o software talvez exiba itens de menu para programas não existentes se você não ficar de olho.

Não obstante, a integração de programas externos realmente torna o software extensível. Por exemplo, os usuários podem usar essa capacidade para acrescentar suas ferramentas de desenvolvimento preferidas.

Gerenciamento de Projeto

O Bluefish realmente brilha quando é preciso gerenciar múltiplos arquivos de um projeto de larga escala. O programa tem uma caixa de diálogo com essa finalidade. Os usuários podem especificar

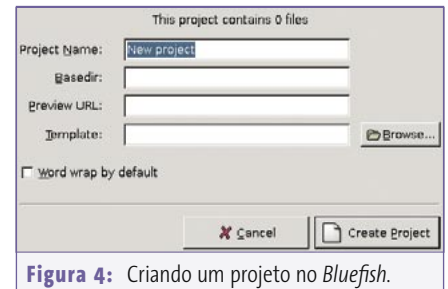


Figura 4: Criando um projeto no Bluefish.

um diretório-alvo para o projeto e atribuir um nome interno do Bluefish; um clique com o botão direito possibilita acrescentar pastas e arquivos à barra lateral.

A caixa de diálogo do projeto também possibilita a definição de um *Diretório Remoto* para o projeto local.

Conclusão

Embora os desenvolvedores digam que a atual versão 1.0 é estável, o software ainda tem um ou dois probleminhas. Por exemplo, o programa travou repetidas vezes enquanto criávamos um projeto e tentamos selecionar um *template* para ele. Certifique-se de fazer uma cópia segura antes de começar a trabalhar com dados de produção.

De acordo com o site oficial do projeto, os desenvolvedores estão trabalhando na incorporação das Diretrizes da Interface Humana do Projeto Gnome (*Gnome Project Human Interface Guidelines*, a conhecida HIG) sempre que possível. Apesar disso, a interface do programa é bagunçada em algumas partes e algumas caixas, como a caixa de configuração da barra lateral, demandam alguma familiaridade.

Em suma, o Bluefish é um editor robusto que pode lhe dar uma mãozinha em seu trabalho de web e programação, assim que você o conheça um pouco melhor. Os blocos de código removem a necessidade de digitação repetitiva e, assim, evitam o erro humano. ■

INFORMAÇÕES

[1] Homepage do Bluefish: bluefish.openoffice.nl

[2] HTML Tidy: tidy.sourceforge.net

Administrando bancos de dados SQL com o knoda

Humm... casa arrumada, hein?

Credits of the images

Mês passado aprendemos como formular pesquisas em múltiplas tabelas no *knoda*. Este mês descobriremos como criar tabelas, formulários e relatórios. Nosso tutorial baseia-se no banco de dados *SQLite 2*, conforme vimos no artigo anterior. Com um pouco de raciocínio, entretanto, você será capaz de seguir as etapas mostradas em qualquer outro banco de dados, como o *MySQL*, por exemplo. O **quadro 1** mostra como configurar o *knoda* para trabalhar com o *MySQL*. Os usuários do SUSE LINUX 9.1 talvez precisem de algum suor para fazer as coisas funcionarem direito: se o KDE tiver sido atualizado para a versão 3.3.2, o YaST pode ter instalado uma versão mais antiga do *knoda*. Nesse caso, basta reinstalar o programa.

Quadro 1: o knoda e o MySQL

O *knoda* trabalha com os bancos de dados *MySQL* e *PostgreSQL*, além do conhecido *SQLite*. Além disso, uma interface ODBC genérica permite acesso a qualquer banco de dados compatível com SQL e ODBC. Para que o *knoda* reconheça esses bancos, é necessário instalar a `hk_class` apropriada para ele. No Debian há versões das `hk_classes` já compiladas para os bancos que já citamos: *MySQL*, *PostgreSQL*, ODBC e *SQLite*.

Se você não for um usuário de Debian (ou derivados como Ubuntu, User Linux ou Knoppix/Kurumin), vai precisar compilar as `hk_classes` novamente com suporte ao banco desejado. É necessário instalar os pacotes de desenvolvimento para cada banco de dados que deseja acessar antes de compilar as `hk_classes`. O *MySQL* precisa do pacote `mysql-devel`, o *PostgreSQL* do `postgres-devel` – e assim por diante.

Depois de compilar as `hk_classes` e o *knoda*, a janela de seleção de drivers mostra

quais bancos de dados as `hk_classes` foram capazes de encontrar. Deve haver uma opção para o *MySQL*. É possível que o *MySQL*, se instalado, já esteja rodando em sua máquina desde o boot. Se não estiver, digite o seguinte comando (como root):

```
/etc/init.d/mysql start
```

Deve aparecer uma linha informando *Done* ou *OK* quando o banco de dados for executado com sucesso. O SUSE LINUX normalmente ordena ao usuário que troque a senha da conta root no *MySQL*. Para isso, digite `mysqladmin -u root password nova_senha`. Se uma senha não for definida, será possível iniciar uma sessão de uso do servidor *MySQL* sem que suas credenciais sejam verificadas. Para iniciar uma sessão no *MySQL*, escolha a opção correta dentro da janela de drivers do *knoda*. Na tela seguinte, informe root como usuário (*User*) e clique em *OK*. Para acessar um servidor *MySQL* pelo *knoda*, especifique o endereço IP (ou o nome da máquina) do servidor no campo *Host*. O sistema pede usuário e senha nesse caso.

O *knoda*, componente do KDE, é um software bastante intuitivo para administração de bancos SQL. Na parte 2 de nossa série, veremos como criar formulários para que os usuários possam alimentar o banco com dados e elaborar relatórios bonitos para impressionar o chefe.

POR MARCEL HILZINGER

Criando um banco de dados

Inicie o *knoda* e selecione o driver para o *SQLite2* na janela de seleção de bancos. Vá em seguida ao menu *File* e selecione a opção *New | Database*. Informe o nome de seu novo banco de dados, que deve aparecer em *Database (banco de dados – nas versões anteriores o campo chamava-se Current database ou banco de dados atual)*. Se o banco que acabou de criar não aparecer, basta selecioná-lo nesse mesmo menu. Para criar uma nova tabela em seu banco, vá ao menu *File | New | Table*. Uma maneira alternativa seria clicar na aba *Elements* e dar um duplo clique na opção *Tables*. Você pode também clicar com o botão direito em *Tables* e escolher a opção *New*.

Na aba *Table – Unamed* (veja a **figura 1**), clique em *New Field* (novo campo) para adicionar novos campos (ou seja, novas colunas) em sua tabela. Não vamos descrever em detalhes todos os tipos de campos possíveis, pois com isso nos afastaríamos dos objetivos propostos para o nosso tutorial. É suficiente dizer que, se pudermos colocar texto, números inteiros e valores em ponto flutuante já será possível fazer um montão de coisas úteis. O tipo *bool* (Booleano) também é útil para certas aplicações: ele pode guardar valores do tipo *sim/não* ou *verdadeiro/falso*. Alguns bancos de dados, como o *MySQL* e o

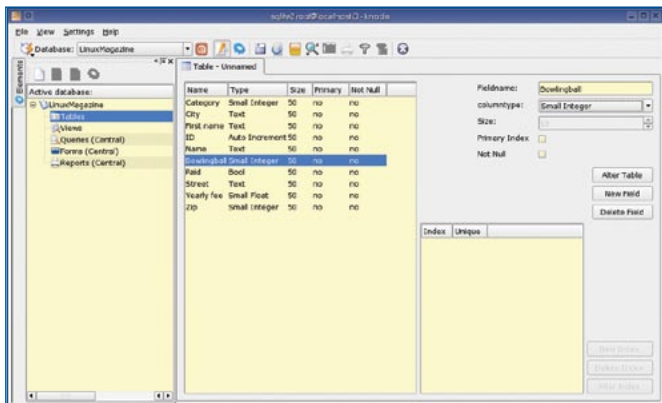


Figura 1: Para o alto e avante: criando uma nova tabela no knoda.

Tabela 1: Exemplo para o clube de boliche.

Nome do campo	Tipo de dado
ID	Auto Increment
Nome	Text
Sobrenome	Text
Endereço	Text
CEP	Small Integer
Cidade	Text
Categoria	Small Integer
Mensalidade	Small Float
Pago	Bool
Bola	Small Integer

PostgreSQL permitem que se coloquem *três* estados nesse campo (por exemplo, *sim*, *não* e *talvez* ou então *1*, *0* e *alta impedância*).

Em nosso tutorial, criaremos um banco de dados para gerenciar as inscrições de membros em um clube de boliche no seu bairro. Para isso, temos que criar os campos listados na **tabela 1**; comece com o campo ID e vá seguindo pela ordem. O campo nunca será zero porque possui um contador automático e também porque será usado como nosso índice principal. Portanto, selecione-o e marque as opções *Primary index* e *Not Null*. Depois de adicionar os campos, vá ao menu *File* e clique em *Save* para salvar a tabela – ou clique no botão *Alter Table*. Na caixa de diálogo *Please enter the tablename* (Por favor, informe o nome da tabela), dê o nome *ClubeBoliche*. Observe que o SQLite não permite que se façam alterações posteriores na tabela após o salvamento, portanto verifique com bastante cuidado se a tabela e seus campos estão do jeitinho que você quer.

Adicione, agora, as categorias desejadas para os jogadores (como Infantil, Infanto-Juvenil, Sub-20, Adultos e Masters) a uma segunda tabela chamada *Categorias*. Em uma terceira tabela, inclua os diferentes tipos de bolas (*LaneMasters*, *Roto-Grip*, *Ebonite* etc). Salve a tabela como *Bolas*. Cada uma dessas tabelas possui apenas duas colunas: uma com o ID e outra com a categoria ou bola desejada.

Os formulários facilitam sua vida

Se você leu a parte 1 de nosso tutorial, publicada na edição anterior, deve se lembrar de que o knoda trabalha em dois modos distintos. Para povoar a nova tabela com dados, simplesmente mude para

o modo de visualização (botão *View mode* na barra de ferramentas) e insira os dados diretamente no “tabelão” mostrado, como se fosse uma planilha comum como a do *Excel* ou do *OpenOffice.org Calc*. Entretanto, há uma maneira muito mais conveniente de se fazer isso: crie um formulário para a entrada de dados. O formulário também permitirá que se façam associações entre as três tabelas. Ou seja, você não precisará saber quais IDs correspondem a que categorias quando as estiver escolhendo. Em vez disso, basta selecionar um item num menu *pull down*.

Para criar um novo formulário, vá até a aba *Elements* e clique com o botão direito em *Forms*, escolhendo *New* no menu de contexto. Se preferir, vá ao menu *File* e escolha a opção *New | Form*. Depois de iniciar o editor de formulários (**figura 2**), é preciso selecionar uma fonte de dados. Para isso, clique no botão com os três pontinhos à direita do menu *Da-*

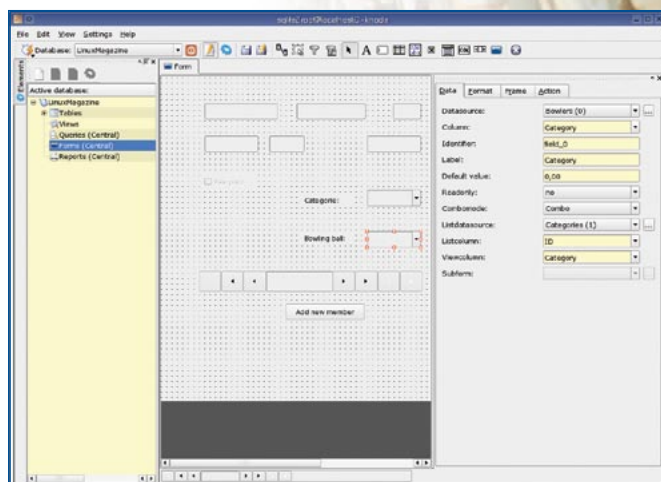


Figura 2: Trabalhando com o editor de formulários do knoda.

tasource. Selecione as três tabelas criadas anteriormente e adicione-as como fontes de dados. Selecione, então, a tabela *ClubeBoliche* no menu *Datasource*.

Ao iniciar o editor de formulários, o knoda automaticamente mostra os elementos necessários para criar um formulário – eles estão representados como botões na barra de ferramentas. Se você não está bem certo do que cada botão faz, basta passar o mouse sobre ele para obter uma dica. Clique no botão *Lineeditfield* e depois no formulário ainda vazio. Isso ordena ao knoda que crie um campo no formulário. É preciso informar ao knoda que tipo de dado esperar nesse campo e como você quer formatá-los. Para isso, clique nas abas *Data*, *Format*, *Frame* e *Actions*. Selecione a tabela *ClubeBoliche* como fonte de dados para o primeiro campo do formulário e atribua a ele o campo *Sobrenome* da tabela. Siga os mesmos passos para adicionar os outros campos, à exceção de *Categoria*, *Bola* e *Pago*. Alterne para o modo de visualização para ver como ficou seu formulário.

O campo *Pago* é do tipo *Booleano*. Ou seja: indica se o membro pagou – ou não – sua mensalidade. Clique no ícone *Booleanfield* na barra de ferramentas, coloque-o no formulário e associe-o ao

campo *Pago* da tabela *TurmaBoliche*. Para ter certeza do que esse marcador faz, digite o texto *Mensalidade em dia* no campo *Label*.

Para os campos *Categoria* e *Bola*, use o elemento *Combobox*. É preciso especificar duas fontes de dados para eles. Primeiro, selecione o campo *Categoria* da tabela *TurmaBoliche* como fonte de dados. Depois, selecione a tabela *Categoria* como *Listdatasource*; defina o campo *ID* como *Listcolumn* e o campo *Categoria* como *Viewcolumn*. Repita esses passos para criar um menu *pull-down* também para os vários tipos de bolas.

Por último, adicione uma barra de navegação ao formulário. Tal barra facilita o ato de mover-se para o próximo registro ou a adição de um registro novo (figura 3). Embora o knoda possua uma barra de navegação para cada formulário, não há como mudar a aparência padrão.

Relatórios

Como era de se esperar, o presidente do clube de boliche pediu ao tesoureiro um relatório dos membros inadimplentes do mês. Você pode usar o Knoda para gerar automaticamente esse relatório e alegrar o chefe. A mágica acontece em *File | New | Report* no menu principal – ou no caminho alternativo pela aba

Elements. Para nosso primeiro relatório, selecionaremos a aba *Bowlers* na caixa de diálogo de fonte de dados.

Clique agora no botão *New Field* na barra de ferramentas e em seguida no campo *Datasection*. Da mesma maneira como no editor de formulários, use as abas do painel à direita para controlar a aparência de seu relatório. Como primeiro elemento, selecione o campo *Sobrenome*. Repita o processo para os campos *Nome* e *Pago*. **Dica:** deixar o campo *Sobrenome* em primeiro facilita a tarefa de agrupar os caloteiros de uma mesma família. Alterne para o modo de visualização para ver seu ainda rudimentar relatório: apenas o nome da família, o primeiro nome e a palavra *TRUE* (se o membro pagou, obviamente) são mostradas. A palavra *FALSE* indica que esse membro é um forte candidato a figurar nas listas do SERASA, SPC e instituições similares.

Obviamente, uma lista muito grande ficaria extremamente atravancada se mantivéssemos esse formato. Uma boa prática é adicionar uma seção para a opção *Pago*. Com isso, o knoda irá mostrar primeiro os maus pagadores, seguidos dos membros honrados que estão em dia com suas obrigações. Para isso, clique no botão *Sections* da barra de ferramentas.

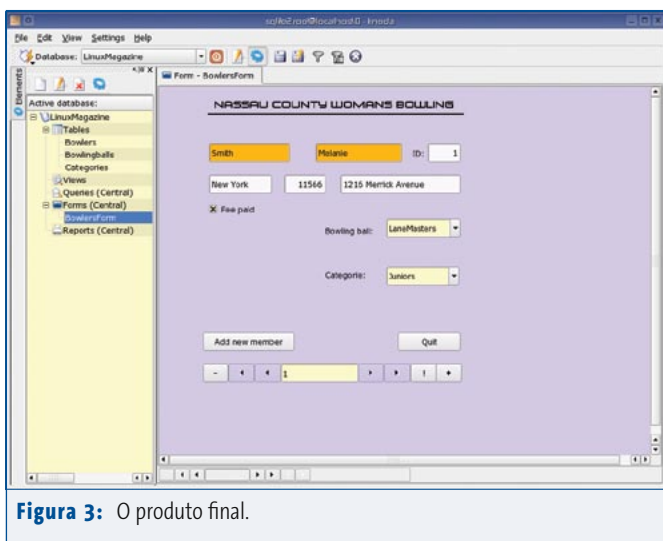


Figura 3: O produto final.

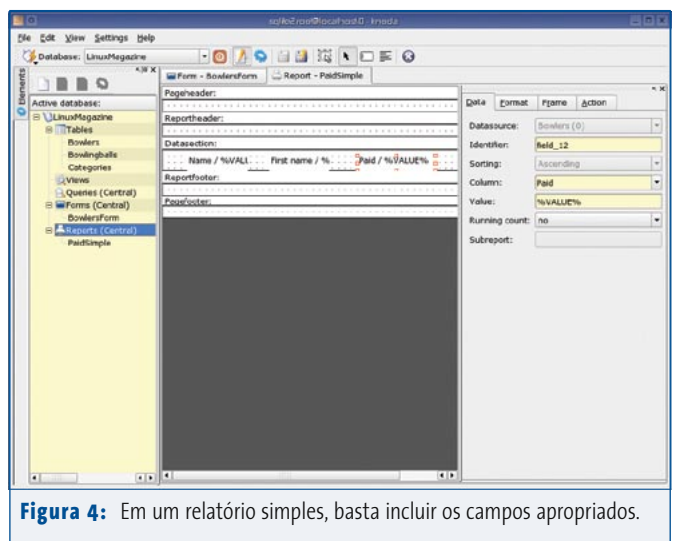


Figura 4: Em um relatório simples, basta incluir os campos apropriados.

Na caixa de diálogo que surge, adicione uma nova seção *Pago* e clique no botão *Exit*. O editor de relatórios possui, agora, dois novos elementos: *Sectionheader:pago* e *Sectionfooter:pago*. Adicione o campo *Pago* no cabeçalho da seção; os demais campos (*Nome* e *Sobrenome*) podem ficar onde estão.

Ainda assim, a lista ainda não é boa o bastante para ser mostrada ao presidente. Para que ele receba um relatório mais sucinto, só com os nomes dos caloteiros que não pagaram a mensalidade, é preciso criar uma consulta SQL, ou *query*. Você pode usar os resultados da consulta como fonte de dados para o relatório. Há outra vantagem nesse método: a lista pode ser classificada de diversas maneiras – por exemplo, alfabeticamente.

Para criar uma consulta, vá em *File | New | Query*, clique com o botão direito no campo de fonte de dados (que deve

estar vazio) e selecione a opção *Add datasource*. Na caixa de diálogo que se abre, escolha a tabela *TurmaBoliche*. Depois, arraste com o mouse os campos que farão parte de nossa consulta e largue-os na matriz. Em nosso exemplo, os campos são: *Sobrenome*, *Nome* e *Pago*.

Defina *=0* como critério para a coluna *Pago* e deixe a opção *Display* como *No*. Alterne para o modo de visualização para enviar a consulta ao banco de dados. Se tudo funcionar como planejado, uma lista com os nomes das famílias seguidos pelos nomes dos membros que não pagaram será mostrada. Para classificar o resultado em ordem alfabética por *Sobrenome*, configure a opção *Sorting* (classificação) para *Ascending* (ascendente) ou *Descending* (descendente).

Agora salve a consulta e crie um relatório em cima dela. Na caixa de diálogo de fontes de dados, indique a consulta que

criamos no campo *Based on*. Adicionar os campos *Sobrenome* e *Nome* na seção de dados e, *voilà*, um relatório instantâneo que qualquer chefe pode entender!

Mais?

Se este tutorial atiçou seu apetite, que tal uma passadinha no site oficial do projeto?

[1] A lista de discussão sobre o knoda [2] é bem animada e traz dicas úteis. E se você está a fim de escovar bits, os tutoriais em [3] são garantia de diversão plena. ■

INFORMAÇÕES

[1] Site oficial do knoda: www.knoda.org/

[2] Lista de discussão, em inglês, sobre o programa e as *hk_classes*: lists.sourceforge.net/lists/listinfo/hk-classes-discuss

[3] Tutoriais diversos, em inglês, sobre as *hk_classes* e o knoda: hk-classes.sourceforge.net/tutorials

Um “sabor” para todos os gostos

Gentoo Linux

O Gentoo Linux nada mais é do que uma distribuição Linux voltada para a otimização e personalização de pacotes. Seu alto desempenho, configurabilidade e uma enorme equipe de desenvolvedores espalhados pelo mundo fizeram desta uma das distribuições mais populares.

POR MARCELO VERIATO LIMA E WILLIAN FERRAZ

Neste artigo vamos apresentar o *Gentoo Linux* aos novos usuários, ensinar a instalar o sistema e desfazer alguns mitos a respeito dessa distribuição. O Gentoo Linux nasceu de uma idéia, a princípio solitária, de Daniel Robbins, que estava descontente com as distribuições que existiam à época, com sua falta de otimizações e gerenciamento de pacotes problemático, entre outras coisas. Porém o principal motivo era que em nenhuma distribuição o usuário podia ter certeza do que estava instalando através de um gerenciador de pacotes. Então, além de projetar o Gentoo, Daniel Robbins também projetou a *Portage*, segundo ele uma ferramenta “mais do que perfeita” comparada com as que existiam na época. Ela foi escrita com foco na flexibilidade do gerenciamento de pacotes.

Portage

Para entender o que é a *Portage* faremos uma analogia com o corpo humano: a *Portage* é comparada às veias que levam o sangue ao coração da distribuição. Através dela é feito todo o trabalho de gerenciamento de pacotes, dependências, busca por informações e atualização de tudo o que está instalado na distribuição. Como o próprio Daniel Robbins disse, “o povo gostou do conceito da *Portage*” e com isso o Gentoo cresceu rapidamente, tornando-se conhecido como a distribuição *From Source* (A partir do

código fonte), o que é um dos aspectos fundamentais e mais importantes do Gentoo Linux.

No início foi complicado ter uma distribuição baseada no código fonte, mas hoje isso é feito sem problemas. Há quem pense que o Gentoo baseia-se apenas no código-fonte, o que não é exatamente verdade: existem vários repositórios não oficiais com pacotes binários para as mais diversas arquiteturas. O projeto da *Portage-ng* (*New Generation*) está caminhando de forma rápida e, logo logo, haverá binários de toda a árvore da *Portage*. O suporte a pacotes binários é muito importante porque eles são largamente utilizados pela comunidade Linux, já que grande parte dos usuários não pode esperar pela compilação de um determinado pacote.

A filosofia do Gentoo se resume em um parágrafo: permitir que o usuário faça o que bem entender com a máquina e desenvolver ferramentas que possibilitem a ele trabalhar de forma agradável e eficiente na instalação de novos pacotes e manutenção do sistema. Nossas ferramentas devem ajudar o usuário a apreciar o Linux e a comunidade do Software Livre. Isso só é possível quando a ferramenta é projetada para refletir e transmitir a vontade do usuário. Se a ferramenta o força a fazer coisas de um modo pré-determinado, ela causará desconforto e estará trabalhando contra o usuário, ao invés de para ele.

Em muitos casos, uma ferramenta parece impor ao usuário “sua vontade”. Com a *Portage* é diferente: absolutamente todas as definições do pacote são atribuídas ao usuário. Outra coisa interessante: a filosofia do Gentoo Linux visa criar ferramentas melhores. Quando uma ferramenta está fazendo o seu trabalho perfeitamente, você não fica muito ciente de sua presença porque a mesma não interfere nas suas decisões, nem o força a interagir com ela quando não quer. A ferramenta serve melhor ao usuário do que o usuário à ferramenta. No futuro, o objetivo do Gentoo é continuar a criar ferramentas quase ideais para as necessidades de todos, ou seja, que possam acomodar tudo de que diferentes usuários possam precisar. Você gosta de uma distribuição ou de uma ferramenta que faz realmente o que você quer fazer? A missão do Gentoo Linux é dar a quem o utiliza essa sensação de saber tudo o que existe instalado no computador e utilizar as ferramentas que preferir.

Guia de instalação

O que muitas pessoas sentem falta no Gentoo é de um instalador amigável. Porém se pegarmos o *Handbook* [1] (o manual oficial de instalação) e o lermos com calma veremos que um instalador não é realmente necessário. O guia é muito didático e mesmo quem nunca trabalhou com Linux já conhecerá um pouco mais sobre o sistema e seus prin-

cipais arquivos apenas com essa leitura. Alguns reclamam que o Handbook “enche lingüiça”, ou seja, enrola demais até chegar ao ponto. Para esses, existe o *Quick Installation Reference* [2], um guia rápido de instalação muito bom.

Desenvolvedores

Qualquer pessoa pode se tornar um desenvolvedor do Gentoo Linux – não é preciso ser um programador nato. Basta testar novos pacotes, informar sobre erros e problemas de compilação ou escrever ebuilds (receitas de compilação em Shell Script) para novos programas. Para isso contamos com um sistema *Bugzilla* muito completo. É só se cadastrar [3] e começar a postar relatórios de erros ou correções. O mais interessante é observar a rapidez com que os bugs são corrigidos e um novo patch ou ebuild é colocado na árvore da Portage.

Documentação

Outra coisa que o pessoal considera excelente é a forma da organização de toda a documentação do Gentoo Linux, dividida por áreas: desde a instalação, documentação para o usuário (Desktop), administração de rede (Serviços), desenvolvimento de ebuilds e como criar e/ou traduzir documentos. Há também um fórum [4] onde os problemas mais comuns, dicas e *how-tos* são encontrados facilmente; existem ainda os canais IRC dentro da rede Freenode.net (irc.freenode.net) onde todos os desenvolvedores se encontram, como o [#gentoo-dev](http://irc.freenode.net) e o [#gentoo-bugs](http://irc.freenode.net). Claro que existem muitos outros, como canais divididos por arquitetura e grupos como o [#gentoo-sound](http://irc.freenode.net), que cuida apenas do conjunto de ferramentas relacionadas a som dentro da árvore Portage. A documentação em Português do Brasil ainda está sendo feita. Quem tiver vontade de contribuir deve mandar um email para o líder do projeto *Gentoo-docs pt_BR*, no endereço anderson@gentoo.org.

Grupo GentooBR

O grupo de usuários *Gentoo Linux Brasil* (GentooBR), do qual fazemos parte, foi formado pelos usuários Otávio Piske e Marcos Vieira há dois anos e meio para atender aos usuários da distribuição dentro do Brasil. Não somos muitos, porém todos se ajudam, conversam e contribuem com a distribuição. Divulgamos o Gentoo Linux em palestras e eventos relacionados ao Software Livre, nos quais muitos usuários nos abordam para conversar sobre o sistema, saber onde podem conseguir-lo e como instalar.

Contamos com um portal [5] onde postamos dicas, how-tos, comentários e demais coisas relacionadas ao Gentoo. Seguimos algumas filosofias, como não ramificar o grupo por região: o GentooBR sempre será o mesmo em qualquer lugar do Brasil. Nosso canal IRC é também o canal oficial do Gentoo Linux no Brasil: [#GentooBR](http://irc.freenode.net) na mesma rede irc.freenode.net.

Instalando o Gentoo Linux

Vamos dar início à instalação, seguindo o passo-a-passo da forma como está no Handbook. A versão usada é a 2005.0, encartada como CD do Mês nesta edição da Linux Magazine Brasil.

A instalação do Gentoo Linux pode ser feita a partir de 3 *Stages* (estágios). Os Stages são como pontos de partida por onde iremos começar a instalação. Veja os prós e contras de cada um:

⇒ **Stage 1** – O usuário terá que compilar toda a distribuição, desde o compilador `gcc` até a `glibc` (processo chamado *bootstrap*), ou seja, toda a base do sistema. Esta opção é ideal para quem quer o máximo desempenho e tem tempo para instalar. Ela permite que você tenha controle total sobre os ajustes de otimização e personalização da distribuição. É apropriada para os *power users* que sabem o que está sendo feito ou para quem gosta de “fuçar”, e permite que você aprenda mais sobre

o funcionamento interno do Gentoo. Entretanto, instalar o Gentoo Linux pelo Stage 1 é demorado e, se você não pretende fazer ajustes finos de compilação, é provavelmente um desperdício do tempo. Não é possível instalar a partir do Stage 1 sem acesso à Internet.

⇒ **Stage 2** – É o Stage 1 pré-compilado para um processador específico e com otimizações feitas pela equipe do Gentoo Linux. Com ele não é necessário realizar o bootstrap, é mais rápido do que começar pelo Stage 1 e você ainda pode personalizar parte do sistema (com o comando `emerge system`). Entretanto, não é possível personalizá-lo tanto como no Stage 1. Essa não é a maneira a mais rápida instalar o Gentoo e você tem que aceitar as otimizações que foram escolhidas para o bootstrap pela equipe de desenvolvedores do Gentoo. Também não é possível instalar a partir do Stage 2 sem acesso à Internet.

⇒ **Stage 3** – Escolhendo o Stage 3 a instalação será a mais rápida, porém as otimizações serão as escolhidas pela equipe de desenvolvimento do Gentoo Linux. Para sermos honestos, os ajustes foram escolhidos com cuidado para realçar o desempenho ao mesmo tempo em que mantém a estabilidade. Utilize esse Stage caso você não tenha uma conexão à Internet durante a instalação. Apesar de ser a maneira mais rápida de instalar o Gentoo Linux, você não pode personalizar o sistema base, pois ele já vem pronto.

Nossa instalação será feita a partir do Stage 1, portanto precisamos de acesso à Internet, de preferência via banda larga (ou muita paciência, caso contrário).

Carregando o LiveCD

Há duas imagens ISO do Gentoo Linux nos *mirrors* oficiais, os CDs *minimal* e *universal*. No *minimal* vêm apenas ferramentas de boot e alguns utilitários; teríamos que fazer o download do stage

a partir do qual desejamos instalar e também de um *snapshot* da árvore da Portage. Já a imagem universal vem com os 3 Stages, um snapshot da Portage e os *distfiles* (código fonte dos programas) para uma instalação sem Internet. Vamos utilizar o CD universal.

Após reiniciar o computador pelo LiveCD podemos escolher quatro imagens do kernel diferentes: *gentoo* (kernel com suporte a *framebuffer*), *gentoo-nofb* (kernel sem suporte a *framebuffer*), *smp* (kernel para máquinas com dois ou mais processadores, com *framebuffer*) e *smp-nofb* (para máquinas “multi-processadas”, sem *framebuffer*). Algumas opções extras também estão disponíveis: tecla [F2] e confira. Caso necessite de alguma delas use-as na forma *kernel opção*, como por exemplo: *gentoo-nofb nodhcp nopcmcia*.

Particionando o disco

Vamos particionar nosso disco como mostrado no Handbook. Digite o comando abaixo:

```
# cfdisk /dev/hda
```

Onde */dev/hda* é o disco *master* na primeira interface IDE. O *cfdisk* é um utilitário de particionamento mais amigável que o popular *fdisk*.

Vamos criar uma partição de “boot” */dev/hda1* com 80 MB e marcá-la como *Bootable* (inicializável). A seguir criamos uma partição de *swap* (também conhecida como memória virtual) */dev/hda2*, tipo *Linux Swap*, com um tamanho igual ao dobro de memória RAM instalada no computador. Pessoalmente, acho que o ideal é seguir essa regra até um limite de 512 MB de *swap*. Por último uma partição */dev/hda3* com o resto do espaço livre no disco. A tabela de particionamento ficará como na [tabela 1](#).

Criando o sistema de arquivos e montando unidades

Particularmente, gosto de mesclar entre os sistemas de arquivo *ReiserFS* e *Ext3*. Vamos criar o sistema de arquivos de nossa primeira partição (boot) como *Ext3*. Digite o comando:

```
# mke2fs -j /dev/hda1
```

Tabela 1: Tabela de Particionamento

Partição	Opções	Tipo part.	Tipo SA	Size (MB)
hda1	Bootable	Primária	Linux	80,00
hda2		Primária	Linux Swap	2x a memória RAM
hda3		Primária	Linux	Resto do disco

A seguir, vamos criar um sistema de arquivos *ReiserFS* em nossa partição raiz:

```
# mkreiserfs /dev/hda3
```

Confirme a alteração teclando [y]. Por último, vamos criar nossa partição de *swap*:

```
# mkswap /dev/hda2
```

E em seguida, vamos ativá-la, com o comando:

```
# swapon /dev/hda2
```

Depois de criar os sistemas de arquivos nas partições, vamos montá-las dentro do diretório */mnt/gentoo*:

```
# mount /dev/hda3 /mnt/gentoo
# mkdir /mnt/gentoo/boot
# mount /dev/hda1 /mnt/gentoo/boot
```



Aviso: Este CD-ROM da Linux Magazine foi testado extensivamente e até pudemos verificar se era livre de qual tipo de software. 2005 h

Configurando a rede

Como estamos instalando o Gentoo pelo Stage 1, necessitamos de uma conexão à Internet, de preferência de banda larga. Vamos explicar os dois casos que podem ocorrer freqüentemente:

1) Caso você possua um servidor DHCP na rede, o Gentoo Linux irá pegar um endereço IP automaticamente. Claro que você também pode configurar o endereço IP manualmente:

```
# ifconfig <eth0> <endereço IP da rede> netmask <máscara> up
```

`eth0` é a interface de rede, `endereço IP da rede` é o endereço de comunicação na rede interna e `netmask` a máscara de rede. Veja:

```
ifconfig eth0 192.168.0.5 netmask 255.255.255.0 up
```

Vamos agora configurar nosso gateway padrão, na forma:

```
# route add default gw <ip do Gateway>
```

Exemplo:

```
# route add default gw 192.168.0.1
```

Depois de configurar o gateway padrão edite o arquivo `/etc/resolv.conf`. No Gentoo Linux o editor de texto padrão ainda é o nano, um editor simples e com poucos recursos, mas que faz o serviço. Digite:

```
# nano -w /etc/resolv.conf
```

E adicione a linha a seguir ao arquivo:

```
nameserver 200.176.2.10
```

Esse endereço IP é um servidor DNS público do Terra Networks. Salve o arquivo teclando `[Ctrl]+[x]`.

Para ter certeza de que sua conexão está funcionando digite:

```
# ping www.meudominiopreferido.com.br
```

2) Caso sua conexão à Internet seja feita usando o protocolo PPPoE (comum em links ADSL), utilizaremos o software `RP-PPPoE` para configurá-la. Digite o comando:

```
# adsl-setup
```

Siga os passos até o final e, logo em seguida, digite

```
# adsl-start
```

Se o resultado for a mensagem *Connected!*, meus parabéns! Você está na Internet!

Instalando o Stage 1 e o snapshot da Portage

Antes de tudo, verifique a data do seu sistema com o comando `date`. Caso ela esteja incorreta vamos acertá-la. `date` recebe como parâmetro uma data no formato MMDDHHMnMnA-AAA, ou seja, **Mês**, **Dia**, **Hora**, **Minuto** e **Ano**, cada um com dois dígitos (exceto ano, com quatro). Por exemplo, a data de *22 de Abril de 2005 às 23:52* é representada por: `date 042223542005`.

Após acertar a data do sistema, vamos entrar no diretório `/mnt/gentoo` (com o comando `cd /mnt/gentoo`) e descompactar o arquivo contendo o Stage 1. Digite:

```
# tar -xvjpg /mnt/cdrom/stages/stage1*
```

Pronto! Agora que o Stage 1 já se encontra no diretório correto, vamos instalar a árvore da Portage:

```
# tar -xvjf /mnt/cdrom/snapshots/portage* -C /mnt/gentoo/usr
```



Configurando as flags para o compilador

A partir de agora iremos configurar as *flags* para o compilador C com os parâmetros de compilação, *USE Flags* e demais coisas. Digite o comando:

```
# nano -w /mnt/gentoo/etc/make.conf
```

A variável *USE* é responsável pelas *USE Flags*; por exemplo, se você deseja o suporte a ALSA em seus pacotes, adicione a entrada "alsa" a essa variável. Outras *USE Flags* podem ser adicionadas, mas na nossa instalação básica isso não fará diferença alguma.

Uma lista das variáveis *USE* pode ser encontrada no arquivo `/usr/portage/profile/use.desc`.

Na variável *CHOST* definimos a arquitetura para a qual o GCC irá compilar os programas. Para máquinas i686 (Pentium II em diante) utilizamos `i686-pc-linux-gnu`, para i386 usamos `i386-pc-linux-gnu`. As flags para as demais arquiteturas estão disponíveis no próprio Handbook.

As *CFLAGS* e *CXXFLAGS* são as mais importantes e devem ser configuradas com cuidado. Na variável *CFLAGS* indicamos a arquitetura de nosso processador (com `-march=`) e flags de otimização para o compilador GCC. Eu particularmente utilizo apenas três flags: `-O2`, `-pipe` e `-fomit-frame-pointer`.

Então, se meu processador for um Intel Pentium 4, utilizo as flags:

```
CFLAGS="-march=pentium4 -O2 -pipe"
```

Essas *CFLAGS* indicam que o sistema será compilado com otimizações específicas (e exclusivas) para processadores Intel Pentium 4 (`-march=pentium4`), procurará otimizar ainda mais o código fonte gerado (`-O2`) e utilizará *pipes* (`-pipe`) para a comunicação entre os estágios da compilação, o que acelera um pouco o processo.

Atenção: Lembre-se de que programas compilados com o parâmetro `-march=<alvo>` irão funcionar apenas na CPU especificada, no nosso caso Pentium 4. Verifique com o comando `cat /proc/cpuinfo` qual o seu processador e indique-o. Para gerar código mais genérico utilize `-mcpu=i686` ou `-mcpu=i386`, dependendo da arquitetura de seu processador, no lugar de `-march`. No caso de processadores Duron, Sempron ou Celeron, o parâmetro `-march` varia de acordo com o core/núcleo do chip.

No site da comunidade Gentoobr.org você encontra uma referência muito boa sobre essas otimizações [6].

Na variável *MAKEOPTS* colocamos `-j2`; ela define quantas compilações paralelas poderemos ter. Para máquinas com dois ou mais processadores informe o número de processadores mais 1. Por exemplo: `-j3` para uma máquina com dois processadores.

Criando o ambiente

O primeiro passo é selecionar um dos servidores espelho (*mirrors*). Digite o comando a seguir:

```
# mirrorselect -i -o >> /mnt/gentoo/etc/make.conf
```

Escolha os mirrors que bem entender, todos são relativamente bons. O próximo passo é copiar o arquivo `/etc/resolv.conf` para `/mnt/gentoo/etc`:

```
# cp -L /etc/resolv.conf /mnt/gentoo/etc/
```

Agora monte o diretório `/proc` sob o diretório `/mnt/gentoo`:

```
# mount -t proc none /mnt/gentoo/proc
```

Agora vem a etapa que tanto esperávamos: criar o ambiente básico para começar a compilação do sistema. Digite:

```
# chroot /mnt/gentoo /bin/bash
```

E exporte as variáveis de ambiente com o comando abaixo:

```
# env-update
# source /etc/profile
```

Parabéns, estamos dentro do ambiente do Gentoo Linux! Claro, ainda faltam alguns passos para terminar a instalação. Sincronize a lista de "pacotes" com as listas oficiais online:

```
# emerge sync
```

O comando anterior utiliza o protocolo *rsync*. Caso tenha problemas com alguma espécie de firewall utilize:

```
# emerge-websync
```

Bootstrap e instalação do sistema

Nesta etapa iremos compilar toda a base do sistema incluindo o GCC, *glibc*, *ncurses*, *zlib* e demais ferramentas necessárias ao funcionamento do sistema. Digite:

```
# cd /usr/portage
# scripts/bootstrap.sh
```

Dependendo de sua conexão com a Internet e da velocidade de seu processador, esse estágio pode demorar um pouco, em média de duas a três horas.

Lembramos que o bootstrap só é necessário para quem vai instalar o Gentoo pelo Stage 1, como é o nosso caso.

Logo após o fim do bootstrap, vamos atualizar as variáveis de ambiente. Digite o comando:

```
# env-update
# source /etc/profile
```

Agora vem outro passo muito importante, a instalação das ferramentas básicas necessárias para que o sistema funcione corretamente:

```
# emerge system
```

Esse processo também demora em torno de duas a três horas (mais se sua máquina for antiga), assim como o bootstrap. Ignore os avisos (*warning*) mostrados na tela. Após o término do processo de compilação, vamos atualizar alguns arquivos de configuração com o comando:

```
# etc-update
```

Configurando o kernel

Antes de tudo escolha o fuso horário (*timezone*), no nosso caso *Brazil-East*:

```
# ln -sf /usr/share/zoneinfo/Brazil/East /etc/localtime
```

Podemos utilizar várias versões do kernel em nosso sistema Gentoo Linux. Vou explicar quatro delas:

- *vanilla-sources* – Kernel 2.4 “padrão” criado pelos desenvolvedores oficiais do kernel Linux, sem patches.
- *gentoo-sources* – Kernel 2.4 com vários patches que corrigem falhas de segurança, melhoram o desempenho e trazem atualizações, preparado pelos desenvolvedores do Gentoo.
- *development-sources* – Kernel 2.6 “padrão” sem patches, assim como o *vanilla-sources*.
- *gentoo-dev-sources* – O mesmo que o *gentoo-sources*, mas na versão 2.6. É o recomendado e o que vamos utilizar. Digite o comando:

```
# emerge gentoo-dev-sources
```

Lembre-se de conferir se o link simbólico `/usr/src/linux` aponta para o diretório correto contendo o código fonte do kernel Linux.

```
# ls -l /usr/src/
lrwxrwxrwx 1 root root 22 Set 30 19:52 linux -> linux-2.6.XX-gentoo-rX
```

Caso o link não esteja correto, execute os comandos abaixo:

```
# rm -rf /usr/src/linux
# cd /usr/src
# ln -s linux-2.6.XX-gentoo-rX linux
```

Troque os “X” pela versão corrente do seu kernel. Para quem sabe compilar o kernel manualmente os passos são os mesmos de sempre:

```
# cd /usr/src/linux
# make menuconfig
```

Configure as opções do kernel ao seu gosto. Lembre-se apenas de preencher alguns requisitos básicos, mostrados a seguir. Um * em frente à uma opção significa que ela está selecionada:

```
Code maturity level options ->
[*] Prompt for development and/or incomplete code/drivers
File systems ->
Pseudo Filesystems ->
[*] /proc file system support
[*] /dev file system support (OBSOLETE)
[*] Automatically mount at boot
[*] Virtual memory file system support (former shm fs)
<*> Reiserfs support
<*> Ext3 journalling file system support
<*> Second extended fs support
```

Caso tenha uma conexão ADSL via protocolo PPP (PPPoE, como no *Speedy*), é necessário habilitar opções extras:

```
Device Drivers ->
Networking support ->
<*> PPP (point-to-point protocol) support
<*> PPP support for async serial ports
<*> PPP support for sync tty ports
```

Compile o kernel digitando:

```
# make && make modules_install
# cp arch/i386/boot/bzImage /boot/kernel-2.6.XX-gentoo-rX
```

Para quem não sabe configurar o kernel há uma alternativa, o *genkernel*, uma ferramenta que ajudará a compilar o kernel automaticamente. Digite:

```
# emerge genkernel
```

Você pode fazer uma compilação totalmente automática do kernel (digite: `genkernel all`) ou acessar um menu de configuração (digite: `genkernel -menuconfig all`). Lembramos que no caso da instalação automática o kernel será totalmente modular.

Para hardware especial ou mais recente há alguns passos a mais. Veja na [tabela 2](#) o que é necessário instalar em cada caso:

Para carregar os módulos, como suporte à placa de rede, áudio ou outros periféricos, digite o comando:

```
# nano -w /etc/modules.autoload.d/kernel-2.6
```

Adicione uma nova linha com a entrada correspondente ao seu módulo; por exemplo, para placas de rede RTL8139 digite

Tabela 2: Módulos extras do kernel

Ebuild	O que é	Comando
nvidia-kernel	Aceleração 3D para placas NVIDIA	<code>emerge nvidia-kernel</code>
nforce-net	Rede on-board nas placas-mãe NVIDIA Nforce2	<code>emerge nforce-net</code>
nforce-audio	Som on-board nas placas-mãe NVIDIA Nforce2	<code>emerge nforce-audio</code>
e100	Placa de rede Intel e100	<code>emerge e100</code>
e1000	Placa de rede Intel e1000	<code>emerge e1000</code>
ati-drivers	Aceleração gráfica para placas ATI	<code>emerge ati-drivers</code>
ati-drivers-extra	Ferramentas gráficas para ATI	<code>emerge ati-drivers-extra</code>

Dica do Geek Slack

Após instalar o sistema instale um pacote chamado *esearch*:

```
# emerge esearch
# eupdatedb
```

Esse pacote faz um “banco de dados” dos pacotes instalados no sistema e após o *esync* (*emerge -sync*) relaciona os pacotes que foram recentemente atualizados na Árvore da Portage. Para maiores informações sobre o uso do *esearch* consulte a página de manual do programa (com o comando *man esearch*).

8139too, para placas Via-Rhine digite *via-rhine* etc. Salve o arquivo e atualize a lista de módulos a carregar:

```
# modules-update
```

Configurações finais

Vamos configurar o *fstab*, arquivo que contém as informações sobre as partições que devem ser montadas automaticamente e os pontos de montagem. Digite:

```
# nano -w /etc/fstab
```

Altere as opções como mostrado na **tabela 3** e salve o arquivo. Vamos definir o *hostname* da máquina. No nosso exemplo, o nome será *larry*. *Larry* é o nome do mascote do Gentoo Linux, *Larry, The Cow*:

```
# echo larry > /etc/hostname
```

Configurando a rede:

Edite o arquivo de configuração da rede com o comando a seguir:

```
# nano -w /etc/conf.d/net
```

Altere os parâmetros conforme sua rede e suas interfaces; altere também o seu gateway. Caso utilize o protocolo PPP para conexão, deixe o gateway comentado. Para fazer com que a placa de rede “pegue” um endereço IP automaticamente durante o boot, digite:

```
# rc-update add net.eth0 default
```

Caso possua mais de uma placa de rede, você deve digitar:

```
# cd /etc/init.d
# ln -s net.eth0 net.eth1
# rc-update add net.eth1 default
```

Agora configure os hosts no arquivo */etc/hosts*:

```
# nano -w /etc/hosts
127.0.0.1 localhost
192.168.0.5 larry larry.cow
```

Para alterar a senha de root digite o comando *passwd*, informe a senha de root e a confirme.

Agora configure algumas opções do sistema no arquivo */etc/rc.conf*

```
# nano -w /etc/rc.conf
```

Neste arquivo você pode alterar o layout do seu teclado no console para *br-abnt2*, seu editor padrão, seu *Display Manager* etc. Em seguida, vamos instalar as últimas ferramentas (*syslog reiser progs* e *cron*):

```
# emerge syslogd reiserfsprogs vixie-cron
```

```
# rc-update add syslogd default
# rc-update add vixie-cron default
```

Se você tem uma conexão ADSL via PPPoE (como o *Speedy*, da *Telefonica*), não esqueça disto:

```
# USE="-X" emerge rp-pppoe
```

Instalando o gerenciador de boot

O gerenciador de boot é responsável por “carregar” o sistema operacional. Vamos ensinar a configurar o *Lilo*, embora também seja possível usar o *GRUB*. Digite:

```
# emerge lilo
```

Edite o arquivo de configuração:

```
# nano -w /etc/lilo.conf
```

E deixe-o como segue:

```
boot=/dev/hda          # Instala na MBR
prompt                 # Menu de seleção do S.O.
timeout=50            # Aguarda 5 segundos
default=gentoo        # Sistema padrão é gentoo
vga=791
# Para quem não utilizou o genkernel
image=/boot/kernel-2.6.XX-gentoo-rX
label=gentoo          # Nome da sessão
read-only
root=/dev/hda3
# Para quem utilizou o genkernel
image=/boot/kernel-2.6.XX-gentoo-rX
label=gentoo
read-only
root=/dev/hda3
append="init=/linuxrc"
initrd=/boot/initrd-2.6.XX-gentoo-rX
```

Salve o arquivo e rode o comando:

```
# /sbin/lilo
```

Hora de reiniciar o sistema e ver se tudo correu bem. Teclê:

Tabela 3: arquivo */etc/fstab*

/dev/hda1	/boot	ext2	noauto,noatime	1 2
/dev/hda2	none	swap	sw	0 0
/dev/hda3	/	reiserfs	noatime	0 1
none	/proc	proc	defaults	0 0
none	/dev/shm	tmpfs	defaults	0 0
/dev/cdroms/cdrom0	/mnt/cdrom	auto	noauto,user	0 0

```
# exit
cdimage ~# cd
cdimage ~# umount /mnt/gentoo/boot
/mnt/gentoo/proc /mnt/gentoo
cdimage ~# reboot
```

Pode ser que o sistema não reinicie com algumas placas-mãe (a máquina travará quando descarregar os módulos USB). Nesse caso aperte o botão de reset que estará tudo bem.

Não se esqueça de tirar o CD do Gentoo do drive de CD-ROM. Se tudo correu bem, seu sistema deverá reiniciar sem problemas. Faça “login” como root e adicione um usuário:

```
# useradd joao -m -G users,wheel,audio,tty
-s /bin/bash
```

Finalmente o Gentoo estará rodando e instalado a partir do Stage 1. Para dominar os comandos básicos, `emerge` e `portage`, recomendamos a leitura dos manuais (`man emerge` e `man portage`)

Instalando pacotes

Como exemplo, vamos instalar o Gnome. Primeiro vejamos as USE-Flags que o pacote disponibiliza:

```
# emerge -vp gnome
```

O parâmetro `-v` mostra as USE-Flags suportadas e `-p` mostra todos os pacotes fundamentais para o funcionamento do Gnome. Caso queira alguma USE-Flag, utilize a variável USE. Por exemplo: `USE="xml2" emerge gnome`. Nesse caso a variável USE colocará suporte a xml2. Caso queira tirar o suporte a xml2, `USE="-xml2" emerge gnome` e assim por diante.

O gerenciamento de pacotes do Gentoo Linux funciona da seguinte forma: visualize as USE-Flags suportadas, passe a variável USE se necessário e retire o parâmetro `-vp` para iniciar a instalação. Para procurar algum pacote dentro da árvore Portage utilize o comando `emerge -s <pacote>`. Por exemplo: `emerge -s apache`. O resultado serão todos os pacotes que contenham a palavra apa-



Figura 01: A comunidade GentooBR [5] está sempre disposta a ajudar os iniciantes.

Tabela 4: Os principais parâmetros do `emerge`

Comando	Resultado
<code>emerge <pacote></code>	Instala um determinado pacote.
<code>emerge -f <pacote></code>	Faz o download do pacote. Para instalar basta remover o parâmetro “-f”.
<code>emerge -s <palavra></code>	Faz uma busca por nome do pacote; suporta expressões regulares.
<code>emerge -S <palavra></code>	Procura na descrição dos pacotes uma determinada palavra; um pouco demorado.
<code>emerge -p <pacote></code>	Visualiza os pacotes que serão instalados.
<code>emerge -vp <pacote></code>	Visualiza os pacotes que serão instalados e também as USE-Flags suportadas.
<code>emerge -u <pacote></code>	Atualiza um pacote.
<code>emerge -uD world</code>	Atualiza toda a distribuição.
<code>emerge -O <pacote></code>	Instala ignorando as dependências. Cuidado!
<code>emerge -C <pacote></code>	Desinstala o pacote.
<code>emerge -i <pacote></code>	Engana a portage fazendo-a pensar que o pacote está instalado. Muito útil quando instalamos um pacote ao modo antigo e queremos que a portage não fique nos avisando sobre as dependências desse pacote.

che no nome. Não se esqueça de ler os manuais do `emerge` e da `portage`. Para sanar dúvidas acesse o canal de IRC do GentooBR, [#gentoobr](#) no servidor [irc.free-node.net](#). Todos os participantes estarão à disposição para ajudá-lo. ■

S.O.S

O CD-ROM que acompanha a Linux Magazine foi testado e, até onde pudemos constatar, se encontra livre de qualquer tipo de vírus ou conteúdo malicioso e de defeitos. Não nos responsabilizamos por qualquer perda de dados ou dano resultante do uso deste CD-ROM ou de software nele incluído. A Linux Magazine não oferece suporte técnico ao conteúdo do CD.

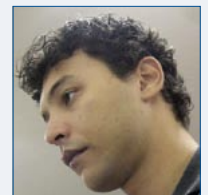
Este CD contém o *Stage 3* do *Gentoo Linux 2005.0* para a arquitetura x86, otimizado para os processadores Pentium III, x86 e Athlon.

INFORMAÇÕES

- [1] Gentoo Install Handbook
- [2] Quick Installation Reference: www.gentoo.org/doc/en/gentoo-x86-quickinstall.xml
- [3] Bugzilla do Gentoo: bugs.gentoo.org
- [4] Fórum do Gentoo: forums.gentoo.org
- [5] Gentoo Linux Brasil: www.gentoo.org
- [6] Parâmetros de otimização: www.gentoo.org/modules.php?op=modload&name=Sections&file=index&req=viewarticle&artid=15&page=1

SOBRE OS AUTORES

William Ferraz (Geek_Slack) é Administrador de Redes e editor da comunidade GentooBR, além de tradutor da documentação



para o site oficial do projeto Gentoo Linux e da Gentoo Weekly Newsletter.



Marcelo Veriato Lima (marcelo_) é Administrador de Redes e editor da comunidade GentooBR.

APT no SUSE Linux

Misturância de pacotes

O *Advanced Packaging Tool* (APT), da distribuição Debian, é o sistema mais avançado (e simples de usar) para gerenciamento e instalação de programas no Linux. Mas e se você não usar o Debian?

Este artigo descreve como usufruir do poderoso sistema APT em outras distribuições baseadas no sistema de pacotes RPM.

POR MIRKO ALBRECHT

O *Advanced Packaging Tool* (mais conhecido pela sigla APT) é um sistema simples mas bastante poderoso para a instalação de software no Debian GNU/Linux. Os pacotes específicos do Debian estão disponíveis tanto em formato binário (já compilado) como em código fonte. Os usuários podem, além disso, acessar um repositório central para atualizar toda sua coleção de programas e bibliotecas pela Internet, com um único comando. As ferramentas de gerenciamento de pacotes lidam automaticamente com quaisquer dependências que precisem ser satisfeitas durante o processo.

Agora os usuários de distribuições baseadas no sistema de pacotes RPM também podem tirar vantagem das benesses providas pelo APT. Usando um sistema SUSE LINUX 9.2 (que distribuimos em nossa sexta edição) como exemplo, este artigo mostrará como administrar a instalação e a atualização de software usando o APT para RPM.

Instalação e Configuração

Se você pretende usar o APT com o SuSE, vá logo contratando uma conexão à Internet via banda larga. O sistema APT sempre procura por programas e atualizações na Grande Rede. Os usuários do SUSE 9.2 precisam instalar dois pacotes:

`apt-0.5.15cnc6-rb.suse092.6.i586.rpm` e `apt-libs-0.5.15cnc6-rb.suse092.6.i586.rpm`. Ambos estão disponíveis em [1]. Se preferir usar uma interface gráfica para gerenciar seus programas, instale também o pacote `synaptic-0.55.3-0.suse092.rb1.i586.rpm`.

Depois de completar essa etapa, é preciso editar e configurar um arquivo chamado `sources.list` (veja **quadro 1: Repositórios**). Esse arquivo especifica quais repositórios devem ser usados

e quais o usuário prefere não usar. Os repositórios, da maneira como o APT os entende, são coleções de pacotes de software – tipicamente classificados por categoria – presentes na Internet ou em CDs. O arquivo `sources.list`, presente no diretório `/etc/apt`, permite que especifiquemos os repositórios usados para atualizar o sistema. Usuários mais cautelosos provavelmente se esquivarão de adicionar repositórios com nomes sus-

Quadro 1: Repositórios

O repositório sempre possui uma área para arquivos binários (*RPMS.nome*) e uma segunda área para os códigos fonte dos programas (*SRPMS.nome*). Apenas a primeira é importante em um sistema para uso no dia-a-dia. Inclua o repositório SRPMS apenas se pretender instalar o código fonte dos programas que usa – ou seja, isso só faz sentido se você for um programador. Um exemplo de lista de repositórios está disponível em [2]; você pode copiar o arquivo para o diretório `/etc/apt` e trocar seu nome para `sources.list`.

Um arquivo `sources.list` mais conservador, que não atualiza todo e qualquer programa de sua máquina para a versão mais recente se parece com este aqui:

```
#
# Repositório criado por: aptate (versão 0.69.0)
# At: Sexta Jan 28 17:16:18 MET 2005
# Mais informações sobre o aptate em: http://apt4rpm.sourceforge.net
#
rpm ftp://ftp.gwdg.de/pub/linux/suse/apt SuSE/9.2-i386rpmkeys base funktronics
update misc mozilla suser-rbossuser-guru suser-gbv packman packman-i686 wine suse-
projects kde3-stable security
```

Se preferir, é possível navegar pelos repositórios e escolher os que melhor satisfazem seus interesses.

peitos como *kde-unstable* (Pacotes instáveis do KDE) ou *kernel-of-the-day* (kernel do dia) a seus arquivos `sources.list`. Além disso, deve-se tomar cuidado com os pacotes do repositório *usr-local-bin*. E, se você está feliz com a configuração de seu ambiente gráfico, deixe de fora o repositório *X.org*.

Trabalhando como *root*, abra o arquivo `/etc/apt/apt.conf.d/gpg-checker.conf` com seu editor de textos favorito e altere o valor `true` da linha `GPG::CHECK` para `false`. Com isso, o APT passa a aceitar pacotes sem verificar sua assinatura. Se isso não for feito, não será possível baixar vários deles, pois infelizmente muitos não possuem uma assinatura digital.

Os usuários mais cautelosos provavelmente não vão querer que o kernel seja automaticamente atualizado. Para impedir atualizações automáticas, abra o arquivo `/etc/apt/apt.conf.d/apt.conf` e adicione à seção RPM em um sistema com kernel 2.6 a linha:

```
Hold {"kernel-default"; "kernel-source";};
```

Se você ainda usa o kernel 2.4, a linha a acrescentar é a seguinte:

```
Hold {"k_deflt"; "kernel-source";};
```

Se configurar o sistema editando diretamente os arquivos de texto não é exatamente a sua praia, use a ferramenta gráfica *Synaptic* para modificar a sua lista de repositórios. Para isso, escolha, no menu *Settings* (*Configurações*) a opção *Package sources* (*Repositórios*).

O APT na linha de comando

O APT é bastante poderoso e pode modificar drasticamente o conjunto de softwares instalados em seu computador. Entretanto, não é fácil entender como a coisa funciona nos bastidores. Nos concentraremos nas opções voltadas para os usuários com um bom conhecimento do YaST e do YOU, a

Quadro 2: APT para versões mais antigas do SuSE

O APT para o SuSE funciona não apenas na versão 9.2 da popular distribuição Linux alemã, como também em qualquer versão mais recente que a 7.3. A arquitetura AMD64 é suportada pelo SuSE 9.x ou posterior.

Os pacotes de instalação do `apt`, `apt-libs` e `synaptic`, estão disponíveis para download em [1]. Simplesmente substitua o número 9.2 na URL pelo correspondente à sua versão do SUSE. Para rodar o APT em versões do SUSE anteriores à 8.2, é preciso ainda instalar o pacote `lua-5.0.2-rb1.i586.rpm`.

O usuário notará ainda algumas diferenças quando comparar o APT nas diferentes versões de SUSE. Por exemplo, as versões 7.3 até 8.1 guardam o arquivo `apt.conf` diretamente no diretório `/etc/apt/`.

ferramenta de atualizações automáticas do SUSE. Os usuários mais "fuçadores" vão se divertir a valer lendo as páginas de manual do APT (digite `man apt` no terminal), onde encontrarão explicações mais aprofundadas sobre as entranhas do bicho.

Depois de configurar o arquivo `sources.list` no diretório `/etc/apt`, digite o comando `apt-get update` em uma janela de terminal. O comando testa a estrutura dos pacotes de software, localiza os dados apropriados nos repositórios listados no arquivo `sources.list` e mostra ao usuário o que pode ser atualizado.

Digite `apt-get install nome_do_pacote` ou `apt-get remove nome_do_pacote` para instalar ou remover, respectivamente, o programa indicado. Os usuários mais valentes podem ficar tentados a atualizar o sistema todo de uma vez usando o comando `apt-get upgrade`. **Nunca** tente tal barbaridade a não ser que saiba exatamente o que está fazendo. Decisões precipitadas ou desinformadas podem levar à instalação de pacotes de origem duvidosa ou instáveis e comprometer a integridade do sistema. Você foi avisado!

Caso realmente queira fazer uma coisa dessas, rode o comando `apt-get -s upgrade` antes de tudo e verifique quais pacotes o APT quer atualizar. (figura 1) Isso vai ajudá-lo a evitar enxaquecas tremendas, já que muitos problemas de dependência são impossíveis de solucionar.

Se, por outro lado, você simplesmente precisar instalar as atualizações oficiais de segurança de sua distribuição – ou, no máximo, pacotes adicionais do repositório oficial – o comando para atualização global `apt-get upgrade` é exatamente o que o doutor receitou. Se preferir sombra e água fresca, talvez queira usar a ferramenta gráfica *Synaptic*.

Usando o Synaptic

Se você ainda não instalou o pacote `synaptic-0.55.3-0.suse092.rb1.i586.rpm`, pode, como *root*, simplesmente digitar `apt-get install synaptic` e esperar. O *Synaptic* foi construído com a biblioteca gráfica GTK. Ah, você não tem o GTK instalado em sua máquina? Não tem problema, o APT vai instalá-la também, já que é uma dependência do pacote *Synaptic*. Se o APT reclamar que não há candidatos

```

ju@junix:~$ sudo apt-get -s upgrade | more
Reading Package Lists...
Building Dependency Tree...
The following packages will be upgraded:
  alsa alsa-devel alsa-tools alsa-tools-gui dar gpg lmlib2 lmlib2-loaders k3b
  kdar kmaildonkey libdar libdv libieee1284 libmpeg2 libmpeg2-libpgg libmpeg2-devel
  libtheora libusb libvorbis libvorbis-devel linekd libxosdplugin
  linekd mjpegtools mjpegtools-mysql connector-java notepadit planner
  sane speak xchat xsane
The following packages have been kept back:
  kaffeine xsimulink.
32 upgraded, 0 newly installed, 0 removed and 2 not upgraded.
Inst alsa [1.0.6-8] (1.0.6-8.pm.0 SUSE 9.2/stable)
Inst gpg [1.2.5-3] (1.2.5-0.2.suse9.2/stable)
Inst lmlib2 [1.1.1-2.2] (1.2.0-8.pm.0 SUSE 9.2/stable)
Inst libmpeg2-devel [1.15-8] (1.1.2-0.pm.0 SUSE 9.2/stable) {}
Inst libmpeg2 [1.1-39] (1.1.2-0.pm.0 SUSE 9.2/stable)
Inst libvorbis-devel [1.0.1-60] (1.1.0-0.pm.0 SUSE 9.2/stable) {}
Inst libvorbis [1.0.1-60] (1.1.0-0.pm.0 SUSE 9.2/stable)
Inst k3b [0.11.15-3] (0.11.20-3.pm.0 SUSE 9.2/stable)
Inst kmaildonkey [0.9.1-247] (0.10-0.pm.1 SUSE 9.2/stable)
Inst kdar [1.2.0-3] (2.0.1-0.pm.0 SUSE 9.2/stable) {}
Inst dar [2.1.4-2] (2.2.1-0.pm.0 SUSE 9.2/stable) {}
Inst libdar [2.1.4-2] (2.2.1-0.pm.0 SUSE 9.2/stable)
Inst libieee1284 [0.2.8-2] (0.2.9-0.pm.0 SUSE 9.2/stable)
Inst libmpeg2 [0.4.0b-20] (0.4.0b-21.pm.0 SUSE 9.2/stable)
Inst libtheora [1.0.1-33] (1.0.1-34.pm.0 SUSE 9.2/stable)
Inst libusb [0.1.8-33] (0.1.10-0.pm.2 SUSE 9.2/stable)
Inst linekd [0.8beta4-2] (0.8.2-0.pm.1 SUSE 9.2/stable)
Inst linekd-defaultplugin [0.8beta4-2] (0.8.2-0.pm.1 SUSE 9.2/stable)
Inst linekd-xosdplugin [0.8beta4-2] (0.8.2-0.pm.1 SUSE 9.2/stable)
Inst libdv [0.103-2] (0.104-0.pm.4 SUSE 9.2/stable)
Inst mjpegtools [1.6.2-54] (1.6.2-54.pm.0 SUSE 9.2/stable)
Inst notepadit [2.7.9-2] (2.7.4-8.pm.0 SUSE 9.2/stable)
Inst planner [0.12-0] (0.12.1-0.pm.6 SUSE 9.2/stable)
Inst sane [1.0.14-5] (1.0.15-0.pm.1 SUSE 9.2/stable)
Inst speak [1.0.4-3] (1.1.7-0.pm.0 SUSE 9.2/stable)
  
```

Figura 1: O comando `apt-get -s upgrade` mostra uma simulação do que aconteceria em uma atualização completa do sistema.

Tabela 1: Principais comandos do APT

<code>apt-get update</code>	Baixa dos repositórios uma lista atualizada dos programas disponíveis e suas versões.
<code>apt-get upgrade</code>	Atualiza automaticamente todos os pacotes instalados que possuam novas versões nos repositórios.
<code>apt-get -s upgrade</code>	Simula uma atualização, para que possamos evitar problemas antes que aconteçam.
<code>apt-get -s install</code>	Simula uma instalação, nos mesmos moldes do comando anterior.
<code>apt-get install nome_do_pacote</code>	Instala o programa indicado e todas as suas dependências.
<code>apt-get remove nome_do_pacote</code>	Remove o programa indicado.

para instalação do GTK, certifique-se de que o repositório `base` está presente em seu `sources.list`. De qualquer forma, o APT *sempre* vai tentar instalar todos os pacotes necessários para você.

Depois de completar a instalação, o Synaptic estará pronto para a luta no *Menu K* (“iniciar”) do KDE – mais precisamente em *System | Configuration | Synaptic Package Manager (Sistema | Configuração | Gerenciador de Pacotes Synaptic)*. Para que isso funcione, é preciso uma versão recente do KDE. Depois de informar a senha do superusuário (root), observe que o painel esquerdo da janela principal do programa está dividido em seções semelhantes às do YaST. O Synaptic lista os programas disponíveis e suas versões no painel à direita. Um quadradinho verde em frente ao nome do programa indica que ele está instalado. Se o marcador possuir além disso um asterisco ou uma estrela, é sinal de que uma versão mais recente daquele programa se encontra no repositório. Para mais informações, consulte *Help | Icon Legends (Ajuda | Legendas dos ícones)*.

A instalação e a atualização de pacotes segue um processo parecido com o do YaST. Clique com o botão direito no pacote desejado e, no menu de contexto, escolha *Mark for installation (Marcar para instalação)* ou *Mark for upgrade (Marcar para atualização)*. Clique no botão *Apply (Aplicar)*. Imediatamente, inicia-se o processo de download e instalação. Para

atualizar a lista de pacotes disponíveis (o equivalente ao comando `apt-get update`), clique no botão *Reload (Recarregar)*.

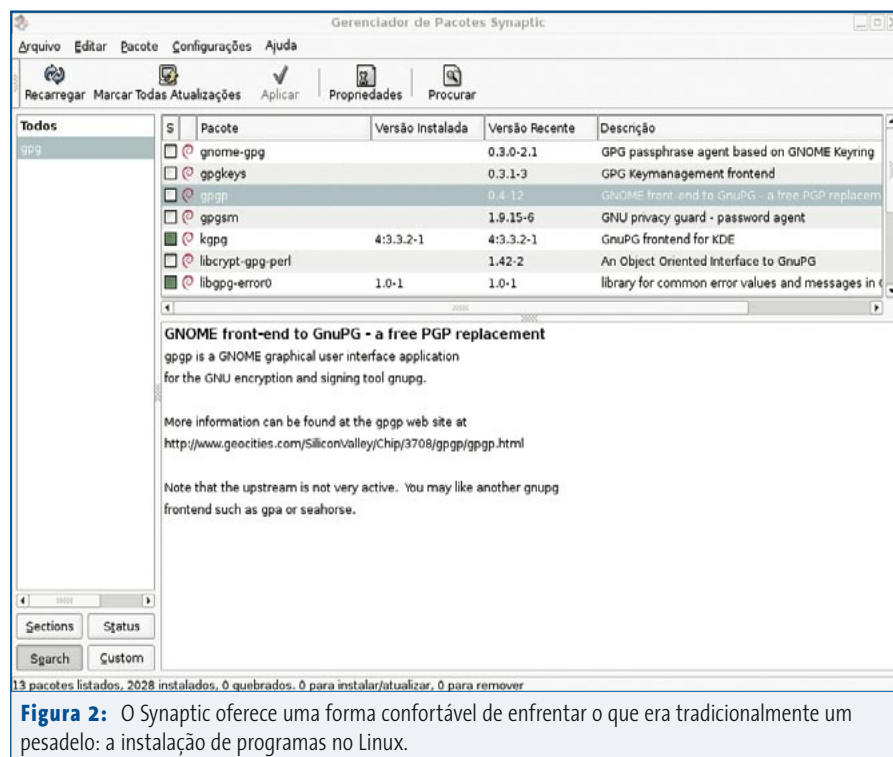
O menu *Settings (Configurações)* possui algumas opções de ajuste bastante auto-explicativas. Você pode querer saber de qual repositório vêm seus programas. Para isso, ative a coluna *Components (Componentes)* na aba *Columns and Fonts (Colunas e Fontes)*, que se encontra no menu *Settings | Configuration (Configurações | Preferências)*. Se múltiplas versões de um mesmo pacote estiverem disponíveis nos diferentes repositórios, o menu *Package*

| *Force version (Pacote | Forçar Versão)* pode auxiliá-lo a escolher a que mais lhe agrada. A opção *Package | Lock version (Pacote | Prender Versão)* permite que a versão de um programa seja “travada”, impedindo que o pacote seja atualizado. Muito útil, por exemplo, quando um programa depende de uma versão específica de um outro programa.

Os pacotes são agrupados por categoria no painel à esquerda. Isso dá aos usuários uma noção mais consistente de como as coisas estão organizadas. Sob a lista existem alguns botões que podem ser usados para alterar o modo de exibição. A opção *Status* permite que os pacotes sejam agrupados por estado: *instalados, não-instalados, atualizáveis* etc. O botão *User defined (Custom em alguns sistemas)* permite controlar as opções de classificação.

Na prática...

Como já dissemos, uma atualização do sistema não é recomendada a não ser que o valoroso guerreiro, digo, o usuário tenha plena convicção do que está fazendo



e seja amigo íntimo do APT já há algum tempo. Os “calouros” devem concentrar-se em atualizações de segurança vindas do repositório oficial ou na instalação de um ou outro programa específico. O sistema APT poupa um tempo razoável mesmo em tarefas relativamente simples como as duas descritas a seguir.

Suponhamos que você seja um feliz usuário do SUSE 9.0 ou 9.1 e está apavorado ante a expectativa de uma atualização radical. Apesar disso, você está louco para ter a última versão do KDE, pois leu na sua Linux Magazine que o ambiente possui inúmeros aperfeiçoamentos, novos programas, uma arte mais bonita e um bom punhado de luzinhas e botõezinhos piscando. A instalação manual pelos pacotes oficiais da SUSE quase o deixou pinel por causa das dezenas de dependências não resolvidas. Hummm.... hora de procurar pelo

KDE no Synaptic. A busca pela palavra *kde* resultou em uma lista de pacotes que podem ser prontamente instalados ou, no mínimo, atualizados. O usuário só precisa selecionar quais os pacotes desejados e deixar que o Synaptic faça o trabalho sujo. Quando clicamos no botão *Aplicar*, o APT começa a lidar com todos os problemas gerados por instalações (ou desinstalações) de pacotes adicionais que o usuário tenha feito antes, bem como a baixar e instalar todas as inúmeras dependências de que o KDE precisa para funcionar.

Da mesma forma, no SUSE 9.2 você pode sentir falta daquele programinha para tocar MP3s que não está nos CDs de instalação e que a SUSE/Novell não incluiu de propósito por questões de licença. Nesses casos, visitar um repositório não-oficial resolve o problema. Antigamente, os usuários precisavam

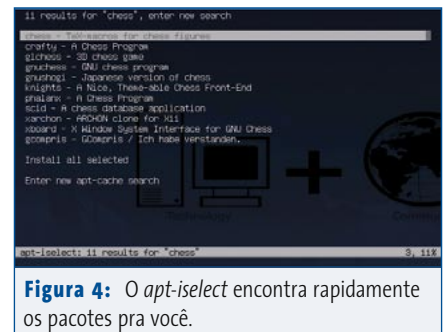


Figura 4: O *apt-iselect* encontra rapidamente os pacotes pra você.

baixar os RPMs do site apropriado [3] e instalá-los à mão. Alguns dos programas – como o *dvd::rip* e inúmeros reprodutores de DVD – obrigavam os pobres usuários a passar a noite em claro pesquisando em fóruns, tutoriais e listas de discussão para descobrir quais pacotes deveriam instalar e em que ordem. Entretanto, se você acrescentar *packman* e *packman-i686* a seu arquivo *sources.list* , instalar o *Xine* , o *MPlayer* e seus comparsas tornou-se sopa no mel. ➡

Dicas, truques, feitiçaria e encantamentos

O APT pode tornar a instalação de software bem mais fácil, mas ainda assim há alguns segredinhos que você deve conhecer para poder operar o sistema de gerenciamento de pacotes de sua distribuição de forma mais segura.

Antes de deixar seu computador fazendo uma atualização radical de madrugada, enquanto você dorme, verifique quais pacotes possuem versões mais recentes. O site Freshrpms [4] é o lugar ideal para isso. Para localizar a página correta, adicione a chave `#suse92` ou `#suse73` à URL – dependendo, obviamente, da versão do SUSE que você possui.

Ocasionalmente, a instalação de um programa qualquer – por exemplo, `abc` – irá falhar, retornando uma mensagem de erro reclamando que “o arquivo `pacote-xyz.so.2` é requerido por `abc`”. É possível dizer ao APT para que instale também o pacote fujão. Basta digitar um comando parecido com: `apt-get install abc pacote-xyz.so.2`.

A ferramenta `apt-iselect` torna as coisas ainda mais fáceis. Combinando a funcionalidade do `apt-cache search` e do `apt-get install`, o `apt-iselect` oferece uma interface simples com o usuário; mesmo em modo texto, é muito fácil de usar. A ferramenta está no repositório `suser-oc2pus`. Para instalar, nenhum segredo: `apt install apt-iselect`. A esta altura o leitor já deve ter pego o “espírito da coisa”.

Uma vez instalado, digite `apt-iselect libgnutls-extra.so.11`. O `libgnutls-extra.so.11` é um arquivo de biblioteca do sistema, e o comando `apt-iselect` nos informou que esse arquivo está no pacote `gnutls`. É possível ver, inclusive, os detalhes do pacote informado, como o número de verificação de integridade (`checksum`), dependências adicionais, descrição, nomes dos arquivos “empacotados”, tamanho do pacote etc. É possível até, se seu coração assim o desejar, proceder à

instalação do dito pacote. Essa é, também, uma boa maneira de encontrar o pacote que faz referência à palavra – por exemplo – “chess” (xadrez). O programa mostra uma lista e, nela, encontramos, entre outros programas para jogar xadrez, o popular `knights`, um `front-end` gráfico bastante atraente para jogar xadrez pela Internet ou contra o computador.

Para remover um pacote, lembre-se de que o comando `apt-get remove nome_do_pacote` irá apenas remover o programa indicado. Quaisquer pacotes instalados para resolver dependências desse programa serão deixados no sistema, ainda instalados e ocupando espaço. Entretanto, se usarmos a opção `-D` (`apt-get -D remove nome_do_pacote`) esses pacotes “sobrantes” também podem ser jogados no latão de lixo. E veja só: isso funciona apenas no `apt-get` para RPMs. No original do Debian essa opção não existe. É a evolução, a criatura superando o criador...

Há ainda um documento ComoFazer (HOWTO) sobre o APT no SUSE em [5]. Ali também podemos encontrar uma vasta quantidade de tutoriais abrangendo um grande número de problemas relacionados ao uso do APT em distribuições baseadas em RPM.

Conclusão

Não é nada difícil instalar, configurar e usar o APT no SuSE. Além de fácil, a preparação do sistema é bastante rápida e, em poucos minutos, o APT está funcionando e pronto para nos dar horas de divertimento sadio. O maravilhoso gerenciador de pacotes Synaptic torna as coisas ainda mais fáceis, levando o APT a usuários bastante inexperientes. Já para a atualização completa do sistema o usuário precisa de um pouco mais de tarimba e cautela devido à “anarquia organizada” dos repositórios. Como qualquer pessoa de bom senso diria, se estiver em dúvida *não faça!*

Este artigo baseou-se no SUSE LINUX para demonstrar como instalar o APT em sistemas que usam o sistema de pacotes RPM. Mas o APT não está, nem de longe, restrito aos usuários do SUSE. A ferramenta está disponível para um sem-número de distribuições baseadas em RPM. É importante lembrar, aliás, que tanto a versão do APT para RPM quanto o Synaptic foram desenvolvidos pela Conectiva (que recentemente se fundiu à Mandrake e hoje se chama Mandriva), num trabalho iniciado pelo lendário Alfredo Kojima, brasileiro que é autor do conhecido gerenciador de janelas WindowMaker. Se você usa o Conectiva Linux, nem precisa se preocupar com nada disso: o APT e o Synaptic são os gerenciadores de pacotes oficiais. Provavelmente isso também acontecerá em futuras versões do Mandriva. A instalação e a configuração dessas ferramentas em outras distribuições baseadas no sistema de pacotes RPM são bastante semelhantes aos passos descritos neste artigo, embora o número de repositórios e alguns outros detalhes possam variar de distribuição para distribuição. Consulte os sites [6] e [7] para mais detalhes. ■

INFORMAÇÕES

- [1] APT para o SUSE LINUX 9.2:
<ftp://ftp.gwdg.de/pub/linux/suse/apt/SuSE/9.2-i386/RPMS.suser-rbos>
- [2] Um arquivo `sources.list` de exemplo:
<ftp://ftp.gwdg.de/pub/linux/suse/apt/SuSE/9.2-i386/examples>
- [3] PackMan, pacotes multimídia para o SUSE:
packman.links2linux.org
- [4] Freshrpms, pacotes não-oficiais fresquinhos:
linux01.gwdg.de/apt4rpm/freshrpms.html
- [5] APT for SUSE How-To:
linux01.gwdg.de/apt4rpm/apt4suse.html
- [6] APT para o Red Hat Linux / Fedora Core:
apt.42h.de/index.en.shtml
- [7] APT para sistemas baseados em RPM:
apt4rpm.sourceforge.net

Todos sabemos que o Linux roda macio em PCs, mas a arquitetura x86 é apenas uma das muitas reconhecidas pelo pingüim. Este artigo mostra como instalar o Linux em uma máquina SPARC da Sun.

POR ARMIJN HEMEL

Novas armas para velhos guerreiros: instalando o Linux em máquinas Sun SPARC

Soltando faíscas

Na época que precedeu o estouro da bolha .com, praticamente todas as empresas de tecnologia possuíam computadores rodando os sistemas da Sun. A plataforma de hardware Sun SPARC também fazia bonito no ambiente acadêmico. Graças a essa abundância de outrora de máquinas SPARC podemos, hoje, comprar uma *SPARCstation* usada a preço de banana – é capaz até que você tenha alguma sem uso enfiada em algum armário ou servindo de apoio para o monitor do chefe.

O Linux já funciona na plataforma SPARC há muitos anos e essa arquitetura é ainda uma das alternativas mais populares para o Linux. De maneira geral, o Linux nas máquinas faiscantes (SPARC é um trocadilho com a palavra inglesa *spark*, faísca) é bastante parecido com o Linux no PC, mas se quisermos que a instalação corra sem problemas temos que prestar atenção às diferenças. Este artigo descreve alguns dos sistemas SPARC [1] suportados pelo Linux, algumas distribuições Linux que funcionam nas SPARCs e muitas das pegadinhas encontradas durante o processo de enfiar um pingüim nas máquinas do solzinho.

Linux ama SPARC

Conseguir uma máquina Sun não é difícil (veja o **quadro 1**). A série *SPARCstation* da Sun vendeu muito bem em sua época e é possível encontrar pilhas de *SPARCstations* 10 e 20 à venda no eBay, por exemplo. No Brasil, lugares como a rua Santa Ifigênia em São Paulo ou o site de leilões Mercado Livre podem ser verdadeiras ilhas do tesouro para máquinas e periféricos, como monitores, se você procurar com vontade. Há também a possibilidade de perguntar em faculdades e empresas se eles não têm alguma máquina que queiram jogar fora.

As SPARCs de 32 bits foram uma das primeiras plataformas para a qual o Linux ganhou uma versão. Muitas distribuições comerciais, como a Red Hat, possuíam “caixinhas” com CDs para sistemas SPARC há mais ou menos seis anos. Como o mercado para o Linux nessa plataforma nunca foi significativo, o desenvolvimento comercial praticamente parou. Hoje em dia todas as distribuições que possuem uma versão para SPARC são desenvolvidas pela comunidade. Algumas das distribuições Linux para SPARC que encontramos são:

- Aurora Linux: as primeiras versões são baseadas no Red Hat Linux 7.3, mas algum esforço está sendo feito para sincronizar a versão atual com o último Fedora Core. Possivelmente os dois projetos se fundirão no futuro.
- Debian GNU/Linux: a mais longeva distribuição para SPARC.
- Gentoo Linux (veja uma matéria sobre ele na seção **Tutorial** desta edição)
- Splack: Slackware para o SPARC.

Como era de se esperar, todas essas distribuições vêm com as mesmas ferramentas de administração também disponíveis em seus irmãos gêmeos para a arquitetura x86 (*RPM* e *yum* no Aurora, *apt* no Debian, *portage* no Gentoo e o *pkgtools* do Splack). Além dessas distribuições, um punhado de outras menores também podem ser instaladas em máquinas Sun.

As máquinas Sun4, as mais antigas com o processador SPARC, não funcionam com Linux. Máquinas SPARC posteriores funcionam com graus variados de sucesso. Algumas máquinas só funcionam em modo monoprocesso, mesmo que possuam mais de um processador instalado – trocando em miúdos, esqueça o kernel SMP para elas. Há famílias inteiras que

funcionam mas que possuem um modelo ou outro, a ovelha negra, com problemas de compatibilidade. Por exemplo, a família *SPARCstation 5* funciona sem qualquer problema, com exceção dos modelos com processador de 170 MHz. A razão para isso é simples: a Sun usou um processador de outro fabricante nesses modelos.

Muito do desenvolvimento atual está sendo feito usando, como “vítimas”, máquinas UltraSPARC mais novas, que possuem barramento PCI para conexão dos

periféricos. Por causa disso, as máquinas SPARC de 32 bits com barramento SBus às vezes “congelam” por instantes. Os kernels mais novos nem sempre colaboram quando instalados nessas máquinas e é preciso testar várias versões até que se encontre uma que resolva o problema. Na outra ponta da corda, os servidores de alta capacidade e última geração são muito caros e, portanto, nenhum *kernel hacker* tem acesso a uma delas para poder escrever código e testar. Por conta disso, o Linux

provavelmente nunca será portado para tais máquinas, embora haja relatos de que alguns loucos conseguiram dar boot no Linux em alguns servidores de média capacidade, como o *Enterprise 4500*.

Instalando Linux no SPARC

Testamos o Aurora SPARC Linux em uma máquina Ultra 10, que tem um processador UltraSPARC Iii e trabalha com discos rígidos IDE, além de possuir um barramento PCI. Por tudo isso, é um dos

Quadro 1: Encontrando uma máquina Sun

Nem toda máquina Sun possui um processador SPARC dentro dela. As máquinas Sun anteriores à versão 4 eram baseadas no processador 68000 da Motorola (o mesmo usado nos primeiros Macs, Amiga e no console Megadrive, da Sega). Umhas poucas, ainda, vieram com processadores Intel 386. Se você quiser rodar um sistema operacional livre nas máquinas com o processador 68000, o NetBSD será sua melhor escolha: ele possui versões tanto para as máquinas *Sun2* como para *Sun3*. Essas máquinas são da década de 80 e praticamente inúteis para os padrões de hoje. A Sun também fabricou algumas máquinas com processadores x86 na linha *Cobalt* e recentemente optou pelo processador AMD Opteron para algumas de suas linhas de servidores e estações de trabalho.

O processador SPARC vem sendo usado desde 1987, quando a Sun introduziu a quarta geração de suas máquinas – conhecidas como *Sun4*. Alguns dos modelos de *Sun4* estão listados na **tabela 1**.

A primeira geração de máquinas *Sun4* usava o barramento VMEbus, também usado nas antigas máquinas com processador Motorola. Logo depois a Sun começou a implementar o barramento SBus, usado desde então em todas as máquinas SPARC. Algumas delas possuem também um barramento MBus para instalar CPUs adicionais no sistema.

Algumas das UltraSPARC mais antigas, como os modelos *Ultra 1*, *Ultra 1e* e *Ultra 2*, usavam o SBus para a instalação de placas de expansão, módulos de memória proprietários da Sun e discos SCA. A Sun manteve essa tecnologia para a maioria dos servidores, com as novas máquinas de alto desempenho usando tecnologias de ponta como o FibreChannel e a possibilidade de conectar e desconectar placas PCI sem desligar o computador. No mercado de estações de trabalho, entretanto, a Sun adotou a arquitetura PC.

As estações de trabalho *Ultra 5* e *Ultra 10* foram as primeiras a adotar comercialmente o barramento PCI, discos rígidos IDE e conectores VGA para monitores – alguns modelos faziam tudo isso nativamente, outros precisavam de uma placa de expansão.

Desde 1996 o processador UltraSPARC é usado em praticamente todas as máquinas, com a notável exceção da fracassada *JavaStation*, a tentativa frustrada da Sun de impingir uma estação de trabalho barata ao mercado, baseada no antigo modelo das máquinas de grande porte – ou seja, onde todo o processamento é feito no servidor. É o que chamamos hoje de *Thin Client* ou *Terminal Leve*. Todas as encarnações do *JavaStation* usavam o antigo processador SPARC (não o UltraSPARC) instalado em uma placa de PC comum, com memórias de PC e *slots* PCI.

Para tornar nossa jornada ainda mais interessante, nem todos os chips SPARC foram fabricados pela Sun. Os padrões para sistemas SPARC são mantidos por uma empresa independente, a SPARC International. Os documentos descrevendo os padrões podem ser baixados gratuitamente e implementados sem que se pague *royalties* para a SPARC International ou a Sun. Há um chip certificado como compatível com o SPARCv8, projetado pela ESA, cujo projeto foi liberado sob a licença LGPL. Todos os documentos relativos a esse processador, chamado de LEON SPARC, podem ser baixados do endereço www.gaisler.com.

No Japão, a Fujitsu fabrica suas próprias versões de SPARC e UltraSPARC. Recentemente, a Sun e a Fujitsu assinaram um acordo de cooperação em que se propõem a trabalhar juntas nas próximas versões da tecnologia SPARC. Há ainda um número razoável de pequenas empresas que fabricam clones de UltraSPARC, como a Solair e a Tadpole (www.tadpolecomputer.com), criadora dos lendários notebook UltraSPARC. Essas máquinas são, internamente, idênticas a uma máquina Sun.

Muitas vezes vemos referências a coisas como *sparcv8* ou *sparcv64*. O primeiro descreve a versão da arquitetura SPARC. Há três versões que podemos encontrar por aí atualmente:

- *sparcv7*: versão 7 da arquitetura SPARC, 32 bits
- *sparcv8*: versão 8 da arquitetura SPARC, 32 bits
- *sparcv9*: versão 9 da arquitetura SPARC, 64 bits

As novas versões possuem mais recursos dos quais um programa pode tirar vantagem enquanto estiver em execução. Muitas distribuições Linux são compiladas para sistemas *sparcv7*, mas alguns pacotes, em particular o OpenSSL, podem se beneficiar de opções de otimização para os novos processadores informadas durante sua compilação. Algumas distribuições, como o Aurora, distribuem pacotes pré-compilados do OpenSSL distintos para processadores *sparcv8* e *sparcv9*.

Termos como *sparcv32* e *sparcv64* não descrevem nenhum processador em particular, mas sim um ambiente de software compilado para uma determinada arquitetura, ou seja:

- *sparcv32*: é um ambiente de 32 bits rodando num processador SPARC
- *sparcv64*: é um ambiente de 64 bits rodando num processador SPARC

É bastante provável que um sistema *sparcv32* possa rodar sem problemas em um processador SPARC de 64 bits. Na verdade, essa é a opção padrão nas distribuições Linux para a plataforma.

modelos em que o Linux funciona melhor. A instalação do Aurora na Ultra não é nem um pouco mais complicada do que a instalação de um Red Hat em um PC comum – afinal, o instalador faz todo o “serviço sujo” pra você. Mas há algumas diferenças fundamentais entre as duas arquiteturas que devemos considerar para que tudo dê certo, mais ainda quando as coisas vão para o brejo.

Uma das diferenças entre a máquina da Sun e um PC comum é a maneira como os discos rígidos são particionados. Num PC podem haver até quatro partições primárias em um disco, uma das quais pode ser uma partição estendida que aloja mais partições lógicas dentro de si.

Um disco num sistema SPARC é tratado de maneira ligeiramente diferente. Podem haver até oito partições em um disco. A terceira partição representa o disco inteiro. Há também restrições com respeito à disposição de partições – por exemplo, uma partição que usa seu setor zero (como a partição de *swap*) não pode estar localizada no cilindro zero, pois isso destruiria o rótulo do disco todo. Além disso, se quisermos transplantar um disco rígido de um PC com Linux para usar em uma SPARC com Linux, é preciso se certificar de que o kernel consegue manipular os rótulos de disco no padrão do PC – coisa que não é configurada por padrão.

A **listagem 1** mostra uma configuração de exemplo para os discos em uma SPARC com Linux.

Ao iniciar o Linux a partir de um CD num sistema com interfaces SCSI, como as Ultra 1, certifique-se de que o drive de CD pode trabalhar com blocos de 512 bytes. O padrão de muitos drives é trabalhar com um tamanho de bloco de 1024 bytes (1 KByte). Nas unidades de CD SCSI de boa procedência, o tamanho do bloco pode ser ajustado com um *jumper*.

A instalação do Linux corre sem erros ou diferenças em máquinas como a Ultra

Quadro 2: Arquitetura de 64 bits

Uma das muitas partes interessantes da arquitetura das UltraSPARC é a possibilidade de rodar executáveis binários de 64 bits compatíveis com o padrão SPARC. Como a maioria das arquiteturas de 64 bits atuais, o processo não é “puro-sangue”, pois as UltraSPARC também podem rodar binários SPARC de 32 bits. Obviamente isso é proposital, para que os clientes da Sun consigam rodar seus aplicativos antigos em seus novos computadores de 64 bits.

Ainda hoje, o Linux nas UltraSPARC é um sistema híbrido com partes em 32 bits e partes em 64 bits. O kernel é um sistema real de 64 bits, mas os programas do usuário são, com raras exceções, todos de 32 bits. Essa divisão tem impacto inclusive nas combinações de utilitários para desempenhar algumas tarefas de compilação de programas (por exemplo, o *gcc* usado em conjunto com o *binutils*). É preciso permitir que os programas possam ser compilados tanto em 32 bits como em 64 bits. A menos que você esteja montando sua própria distribuição, não precisa verdadeiramente se preocupar com essa pagagaiada toda; basta usar os programas fornecidos pelo fabricante e pronto.

Em diversos documentos sobre o Linux nas UltraSPARC lemos que é preciso usar o compilador *egcs64* para recompilar o kernel. Isso era verdade há alguns anos; hoje em dia é possível usar o próprio *GCC*, desde que seja uma versão recente. Como usuário final, não é necessário ativar nenhuma opção extra. O processo de configuração do kernel se encarrega de escolher o compilador mais apropriado para você.

Muitos dos programas dos usuários – compilados para 32 bits pela distribuição – se beneficiariam de uma compilação em 64 bits: afinal, são bem mais lentos que seus “gêmeos” mais modernos. Há ainda outra razão para recompilar os programas mais usados para 64 bits: portabilidade.

O mundo dos 64 bits tem regras bastante diferentes do mundo de 32 bits, especialmente quanto ao tamanho dos ponteiros. Em sistemas de 32 bits, como os PCs, o tamanho dos ponteiros nas linguagens C e C++ é de 32 bits. O tamanho de um valor inteiro é, também, 32 bits. Muitas vezes variáveis do tipo inteiro (*integer*) são usadas no lugar dos ponteiros. Mesmo que isso não seja correto do ponto de vista técnico, o compilador C apenas avisa o fato ao usuário, mas não indica o evento como erro. É apenas um aviso.

Nas plataformas de 64 bits, a história não é nem um pouco diferente. Os ponteiros são de 64 bits, enquanto os inteiros ainda “medem” 32 bits – o que é lógico, pois essas unidades de armazenamento de dados (*byte*, *palavra*, *inteiro*) estão padronizadas há muito tempo. Atribuir um ponteiro de 64 bits a uma variável inteira de 32 bits seria o equivalente a “mutilar” os 32 bits mais significativos do ponteiro, o que poderia levar a resultados bem... “interessantes” (Hê! Hê! Hê!...).

Muitas distribuições instalam apenas os pacotes compilados a 32 bits, mas um ambiente de 64 bits pode ser adicionado com pacotes à parte.

Quadro 3: Quebrando ovos

Em informática usamos uma alegoria retirada das *Viagens de Gulliver* para definir a organização na memória dos bytes (conjuntos de 8 bits) quando agrupados dentro de uma variável de vários bytes – como uma variável inteira, que tem 32 bits e, portanto, 4 bytes. No livro de Jonathan Swift, dois povos brigavam entre si a respeito do modo certo de quebrar um ovo. Os *ovogrossenses* (*big endians*) achavam que o ovo deveria ser quebrado em sua extremidade mais grossa. Já os *ovofinenses* (*little endians*) argumentavam que o local apropriado para se quebrar um ovo era a extremidade pontuda.

Em sistemas com a organização *big endian*, o byte mais significativo é armazenado primeiro – ou seja, na posição mais baixa da memória. Em sistemas *little endian*, o byte menos significativo é armazenado primeiro. Todos os chips SPARC são *big endian*, enquanto as CPUs x86 são *little endian*. Em alguns casos isso pode causar problemas. Por exemplo, há programas que usam formatos especiais de arquivos binários para gravar seus dados. Alguns deles esperam que seus bytes sejam organizados numa ordem específica dentro do arquivo. Esses casos são bastante raros – eu mesmo presenciei apenas uma meia dúzia deles, quase todos relacionados a velhos programas para Macintosh como o MacWrite II. Raro, entretanto, não é inexistente, portanto mantenha seus olhos bem abertos.

Listagem 1: Configuração dos discos rígidos para uma SPARC com Linux

```
Disco /dev/hda (Sun disk label): 16 cabeças, 63 setores, 17660 cilindros
Unidade = cilindros de 1008 * 512 bytes
```

Dispositivo	Início	Fim	Blocos	Id	Sistema de arquivos
/dev/hda1	0	102	51408	1	Boot
/dev/hda2	102	16620	8325072	83	Linux native
/dev/hda3	0	17660	8900640	5	Disco inteiro

10. Para máquinas mais antigas, especialmente as que usam o barramento SBus ou um processador de 32 bits, a situação pode ser bastante diferente. Antes de instalar o Linux, verifique com o fornecedor de sua distribuição favorita qual hardware é oficialmente suportado pelo produto. Um excelente ponto de partida é a lista de compatibilidade de hardware do Aurora.

O processo de inicialização

Uma das maiores diferenças entre as plataformas PC e SPARC é o “boot”. O PC usa uma BIOS extremamente simples (simplicidade?) para inicializar o hardware, encontrar o gerenciador de boot no início do primeiro disco rígido e carregá-lo na memória. Nas máquinas SPARC, a *OpenBoot PROM* (OBP) é usada. A OBP possui um conjunto de comandos bastante flexível e poderoso para carregar um kernel na memória (por meio de discos rígidos, unidades de fita, CD, rede), bem como uma porção de comandos para diagnóstico do sistema. Com o nome de *OpenFirmware*, a OBP também é usada em todos os Macs PowerPC produzidos desde 1994, servidores da IBM e placas PegasosPPC.

Para iniciar o Linux, um gerenciador de inicialização especial chamado *SPARC Improved boot Loader* (SILO) é usado. O SILO pode iniciar sistemas Solaris, SunOS e Linux a partir do disco rígido. Há ainda uma variante especial, preparada para redes, chamada TILO, que permite o boot através da placa de rede. O SILO é bastante parecido com o GRUB, já que conhece a disposição de alguns sistemas de arquivos e, ao contrário do LILO, não precisa ser reinstalado toda vez que a configuração é alterada. A configuração do SILO pode ser traiçoeira, mas o manual é bem claro.

Alternativas

Se você achou esse negócio de rodar o Linux em máquinas SPARC muito complicado, há algumas alternativas atraentes. Em primeiro lugar você pode, é claro, instalar o Solaris, sistema operacional desenvolvido pela própria Sun e, por isso, com funcionamento garantido num número maior de máquinas. A Sun lançou o Solaris 10 sob uma licença chamada CDDL (*Common Development and Distribution License*) que, dependendo da pessoa para quem você perguntar, é aberta.

Se quiser uma alternativa que realmente seja aberta, experimente um dos inúmeros membros da família BSD. O NetBSD possui já há bastante tempo uma versão para a arquitetura SPARC. O suporte a SPARC do OpenBSD é bastante festejado, já que muito do desenvolvimento do OpenBSD é feito justamente em máquinas UltraSPARC. As novas UltraSPARC III, entretanto, não são reconhecidas pelo OpenBSD, uma vez que a Sun não teve a boa vontade de disponibilizar as especificações do hardware em termos aceitáveis para a equipe de desenvolvimento do sistema.

Já a versão para SPARC do FreeBSD é bastante recente. Os desenvolvedores se voltaram apenas para as máquinas UltraSPARC, deixando de lado as antigas SPARC baseadas em processadores de 32 bits. Nem todas as máquinas funcionam e, mesmo para as que funcionam, alguns periféricos, como a placa de vídeo, ainda precisam de bastante trabalho. ■

Informações

- [1] Visão geral dos sistemas da Sun: sunsolve.sun.com/handbook_pub/Systems
- [2] Guia de nomes para as CPUs SPARC: www.sparcproductdirectory.com/sparccpu.html
- [3] Lista de hardware compatível com o Aurora Linux: auroralinux.org/cgi-bin/wiki.pl?HardwareCompatibility
- [4] SILO - Gerenciador de boot para máquinas SPARC: www.sparc-boot.org
- [5] Página da Wikipedia sobre ordenação de bytes: en.wikipedia.org/wiki/Endianness
- [6] Debian GNU/Linux para SPARC: www.debian.org/ports/sparc
- [7] Splack Linux: www.splack.org
- [8] Gentoo Linux para SPARC: www.gentoo.org/proj/en/base/sparc/index.xml
- [9] História da arquitetura SPARC: en.wikipedia.org/wiki/Sparc
- [10] LEON, um processador SPARC Open Source: www.gaisler.com/index.html

Tabela 1: Subtipos Sun4

Tipo da máquina	Tecnologia (bits)	Barramento	Exemplos
sun4	32	VMEbus	Sun-4
sun4c	32	SBus	SPARCstation SLC/ELC/IPC/IPX/1/2
sun4d	32	SBus	SPARCcenter SPARCserver
sun4m	32	SBus	SPARCstation 4/5/10/20; SPARCstation LX/ZX/Voyager; SPARCclassic
sun4u	64	Sbus/PCI	Ultra; Blade; Enterprise; Fire; Fujitsu SPARC64

Protegendo seus dados

Segurança de ponta a ponta

Quando os pacotes deixam o seu computador e passam a trafegar em redes públicas, nunca se sabe o que pode acontecer com eles. Sempre que a rede de um conjunto empresarial é baseada em *hubs*, qualquer computador conectado ao mesmo hub que você poderá interceptar todo o conteúdo que passa pela placa de rede do seu micro. Existem softwares (desenvolvidos para fins legítimos!), como o *Ethereal* [1], que facilitam a análise desse tipo de tráfego, com requintes como destacar os logins e senhas, remontar os emails que tenham sido fragmentados no trajeto e até exibir as imagens que estiverem sendo transferidas pelo seu navegador. E isso sem necessidade de acesso ao seu computador – é uma decorrência da maneira como os hubs funcionam (e uma boa razão para substituí-los por *switches*).

Por exemplo: você vai passar alguns dias na casa de um parente que tem Internet compartilhada. Como ter certeza de que suas senhas ou mensagens confidenciais não serão vistas por alguém mal-intencionado que more no mesmo condomínio? E se você precisar transferir um arquivo estritamente confidencial entre duas filiais de sua empresa via Internet, até que ponto você está disposto a confiar na segurança e discrição de todos os provedores pelos quais essa transmissão irá passar?

A solução ideal – e nem sempre possível – para esse tipo de situação costuma ser uma VPN (rede virtual privada). Mas se você tiver acesso a um servidor externo

em cuja segurança confia, o uso dos túneis baseados no SSH pode ser uma boa solução temporária.

Imagine a seguinte situação: você está em um condomínio usando o computador *C* e quer buscar seus emails no servidor pop.provedor.com.br (identificado pela letra *P*). Se você fizer esse acesso pela maneira direta tradicional, sua senha e suas mensagens poderão ser interceptados por outros computadores na mesma rede local.

Mas se você tiver acesso remoto (via SSH) a um outro computador *S* que considere seguro (por exemplo, um computador na sua empresa, ou mesmo em sua casa, permanentemente ligado à Internet), poderá criar o que chamamos de um *túnel SSH* entre a máquina *C* e a máquina *S* e acessar seu email através desse túnel. Assim, os dados trafegarão criptografados na sua rede local (fluxo número 1 do diagrama) e só serão decodificados entre *S* e *P* (fluxo 2). O servidor *P* não precisa de nenhuma configuração especial e nem saberá que parte do tráfego ocorreu de forma incomum.

Existem muitas formas de se obter esse efeito, mas a mais comum é através do próprio comando `ssh`. Supondo que o endereço do computador *S* seja 200.201.202.203 e que seu login seja *fulano*, você normalmente iria se conectar a ele pelo SSH usando o comando `ssh -l fulano 200.201.202.203`, certo? Mas, para formar o túnel, vamos agregar alguns parâmetros extras. Se você digitar (como root) o comando `ssh -l fulano -L 110:pop.provedor.com.br:110 200.201.202.203`,

Quando você não tem confiança na rede pela qual seus dados trafegam, os túneis SSH podem ser uma solução.

POR AUGUSTO CAMPOS

vai ser aberta uma sessão SSH comum, mas além disso a porta 110 local de *C* passará a ser a entrada do túnel criptografado que passa por *S* (onde é decodificado) e termina na porta 110 (normalmente usada pelos servidores de email POP) do servidor *P*. Essa configuração perdura enquanto a sessão SSH estiver aberta.

Parece complicado, mas o efeito é bastante simples: agora basta abrir o seu programa de email favorito e informar a ele que seu servidor POP deixou de ser o pop.provedor.com.br e passou a ser 127.0.0.1. Na próxima vez que você tentar acessar seu email, abra o túnel e os dados trafegarão criptografados por sua rede local, dificultando qualquer interceptação.

A mesma técnica pode ser usada para outros serviços de rede, basta saber suas portas de acesso [2] e modificar o exemplo acima. Quando se trata de túneis, o limite é a sua criatividade. Eles podem ser usados até mesmo para ultrapassar bloqueios impostos por alguns provedores. ■

INFORMAÇÕES

[1] *Ethereal*: www.ethereal.com

[2] Lista de portas: www.iana.org/assignments/port-numbers

[3] Artigo da O'Reilly: www.oreillynet.com/pub/a/wireless/2001/02/23/wep.html

[4] Quebrando Firewalls com OpenSSH: soutpnuts.sourceforge.net/sshtips.htm

[5] Firewall Piercing mini-HOWTO: www.faqs.org/docs/Linux-mini/Firewall-Piercing.html

Alta disponibilidade para VPNs

Caminhos alternativos

O IPsec impede que muitos dos truques espertos oferecidos pelos produtos de alta disponibilidade funcionem. Mostraremos este mês uma solução que pode ser usada como um caminho alternativo para conexões IPsec.

POR JOHANNES HUBERTZ

Os administradores de sistema insistem em usar conexões de rede que possuam um segundo caminho de “backup”, que entra em cena automaticamente caso a conexão principal caia. Mas se usarmos uma VPN com IPsec para proteger o tráfego que deve passar pela Internet, a conexão reserva vai precisar de alguns cuidados especiais.

A razão de ser desses cuidados é o fato de o IPsec [1] precisar de um endereço IP consistente e imutável em ambos os pontos de entrada da VPN. Por isso, se a conexão for comutada para um túnel diferente, o endereço IP deve ser reconfigurado nos dois novos pontos de acesso – caso contrário, qualquer conexão existente será assassinada. O *Border Gateway Protocol* (BGP ou protocolo de gateways de fronteira [2]) oferece uma maneira eficaz e confiável de administrar um conjunto de números IP fornecidos por diversos provedores. Infelizmente, a maioria dos contratos de serviço de conexão à Internet proíbe terminantemente o uso do BGP em seus links.

Como paliativo, muitos administradores fazem tudo “à mão”, sem qualquer automação – quando o ruim vira péssimo, eles trocam manualmente os cabos no painel de conexões para que o sistema use a conexão auxiliar. Não é exatamente o que podemos chamar de tecnologia de ponta em plena “Era da Informação”. Em vez disso, seria preferível que os dispositivos de rede detectassem as falhas em uma

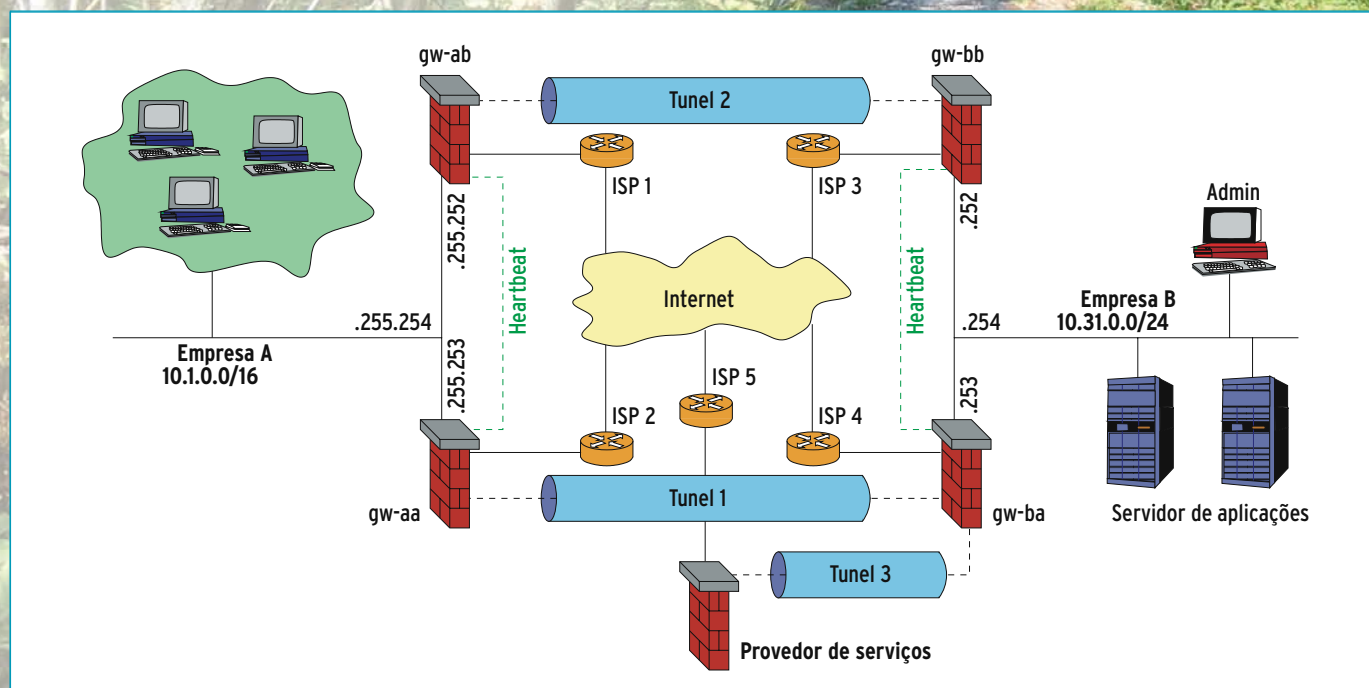


Figura 1: Nesta ilustração, a rede de um cliente está conectada a um prestador de serviços terceirizado por meio de uma VPN. A conexão usa dois caminhos alternativos – se necessário, o Túnel 2 toma o lugar do Túnel 1. Há ainda um provedor de serviços de TI conectado, mas sem alta disponibilidade.

conexão e fizessem automaticamente a comutação para a linha auxiliar. O sistema ideal poderia também gerar automaticamente as configurações para ambos os pontos de acesso, armazenando-as em um local central.

Para gateways que funcionam como firewalls e usam IPsec, a configuração central é o que podemos chamar de tecnologia de ponta. No Linux, o SSPE (*Simple Security Policy Editor* ou editor simplificado de políticas de segurança [3]) pode cuidar dessa tarefa. Entretanto, a solução de alta disponibilidade (ou HA, de *High-Availability*) ainda não é compatível com SSPE.

Linux-HA

O projeto Linux-HA [4] é dedicado a soluções de alta disponibilidade em servidores Linux. Esse software permite que os administradores montem uma VPN

de alta disponibilidade que comute automaticamente e rapidamente da conexão principal para a de reserva – e tudo isso sem usar o BGP. Para implementar essa solução, você precisa de dois túneis paralelos e independentes. Um deles sempre será usado em um determinado momento. Cada túnel, em cada rede, possui seu próprio ponto de acesso, que serve como *gateway* para a rede local. O Linux-HA implementa a reconfiguração automática de endereços IP e pode, com isso, auxiliar nesse “pepino”.

Ambos os nós HA possuem um endereço IP individual e um endereço IP compartilhado, que é usado por apenas uma das máquinas em um dado momento. O mecanismo pode “hospedar” servidores (Web, Mail ou mesmo o bom e velho Doom) que se conectam através do IP compartilhado. O serviço roda em ambas as máquinas e “escuta” em cada

endereço IP. Entretanto, as solicitações de conexão vindas de outras máquinas chegam apenas pelo IP compartilhado. Isso permite que o Linux-HA atribua o endereço externo a uma segunda máquina em caso de emergência; os usuários não perceberão nada – nem que a máquina principal caiu, nem que o endereço externo mudou, já que para eles o endereço que vale é o compartilhado.

Um cabo serial assegura que o *heartbeat* (“batimento cardíaco”) das duas máquinas seja monitorado. O *heartbeat* é uma parte importante do subsistema de alta disponibilidade Linux-HA. Os computadores envolvidos verificam, periodicamente, a disponibilidade de seus parceiros. Se um computador entrar em parafuso (ou se alguém chutar o cabo de rede) o computador parceiro “adota” o endereço IP pertencente ao primeiro computador. Isso gera *broadcasts* via

protocolo ARP usando o endereço IP compartilhado e o endereço MAC da segunda máquina, a que assumiu o controle – pense no procedimento todo como se fosse um ataque de *ARP-Spoofing* ou *ARP-Poisoning*, só que usado com um propósito legal (quem diria que tecno-

logia inventada por bandidos seria empregada por nós, que estamos “do lado claro da força”).

Além do APR-Spoofing, a máquina define um *alias* (apelido) para a interface de rede. Quando o primeiro nó (o que caiu) voltar à vida (ou seja, seu “coração”

voltar a bater) o protocolo *heartbeat* assegura que o segundo servidor desabilitará o alias da interface. Enquanto isso, o servidor principal levanta sua placa de rede e faz um novo *broadcast*, via protocolo ARP, para reassumir o controle sobre aquele endereço IP.

Listagem 1: Supervisor VPN para alta disponibilidade

```

01 #!/bin/bash
02 # Supervisor HA VPN no gateway gw-aa
03
04 # A outra extremidade do túnel
05 TARGET="gw-ba"
06
07 # Número de segundos entre “pings”
08 TIMEOUT=1
09
10 # Espere até que o tempo decorrido seja
11 # MAXFAIL * TIMEOUT antes de habilitar a troca de túneis
12 MAXFAIL=5
13
14 # Espere até que o tempo decorrido seja
15 # HYSTERE * TIMEOUT depois que a máquina principal
16 # entre novamente em operação para, só então, colocar
17 # os túneis na condição inicial
18 HYSTERE=180
19
20 # Considera sem erros o início de operação
21 FAIL=0
22
23 VERBOSE=""
24
25 ACTION_FAIL_START="/root/bin/HA-VPN-action-script start"
26 ACTION_OK_AGAIN="/root/bin/HA-VPN-action-script stop"
27
28 PING=/usr/bin/echoping
29 LOG="/usr/bin/logger -t HA-VPN"
30
31 math () {
32     eval echo "\${($*)}"
33 }
34
35 echo "`date +%Y%m%d%H%M%S` `basename $0` iniciando" | $LOG
36
37 while :
38 do
39     VAL=`$PING ${VERBOSE} -u -t $TIMEOUT -s 5 ${TARGET} 2>&1`
40     ERROR=$?
41     if [ $ERROR -gt 0 ] ; then
42         echo "$DAT $ERROR $FAIL $VAL" | $LOG
43         # Ocorreu um evento de expiração de tempo
44         if [ $FAIL -lt 0 ] ; then
45             # Outro erro durante a fase de recuperação
46             FAIL=`math $MAXFAIL + 1`
47         fi
48         if [ $FAIL -eq $MAXFAIL ] ; then
49             # Iniciar troca para o Túnel 2
50             :
51             FAIL=`math $FAIL + 1`
52             echo "$DAT iniciando troca de túneis: ${ACTION_FAIL_2
53             START}" | $LOG
54             ${ACTION_FAIL_START}
55         else
56             if [ $FAIL -lt $MAXFAIL ] ; then
57                 FAIL=`math $FAIL + 1`
58             fi
59         else
60             # “Ping” detectado com sucesso
61             if [ $FAIL -gt $MAXFAIL ] ; then
62                 FAIL=`math 0 - $HYSTERE `
63             fi
64             if [ $FAIL -le $MAXFAIL -a $FAIL -ge 0 ] ; then
65                 FAIL=0
66             fi
67             if [ $FAIL -lt 0 ] ; then
68                 # Espera pelo período de “histerese” antes de voltar ao 2
69                 Túnel 1
70                 echo "$DAT $ERROR $FAIL $VAL" | $LOG
71                 FAIL=`math $FAIL + 1`
72                 if [ $FAIL -eq 0 ] ; then
73                     # Restaura operação normal
74                     :
75                     echo "$DAT novamente em operação normal: ${ACTION_2
76                     OK_AGAIN}" | $LOG
77                     ${ACTION_OK_AGAIN}
78                 fi
79             fi
80             #echo "$DAT $ERROR $FAIL $VAL" | $LOG
81             sleep $TIMEOUT
82             done
83             # A instrução abaixo nunca é atingida.
84             # Está aqui somente para fins de documentação
85             exit 0

```

Listagem 2: Script de acionamento da VPN de alta disponibilidade

```

01 #!/bin/bash
02 # Script de acionamento HA-VPN
03
04 #VERBOSE=-v
05 VERBOSE=""
06
07 NAME=`basename $0`
08 LOG="/usr/bin/logger -t HA-VPN"
09
10 PARAMETER_FAULT=0
11
12 if [ $# -ne 1 ] ; then
13     PARAMETER_FAULT=1
14 else
15     PARAMETER=$1
16     case $PARAMETER in
17     start) ;;
18     stop) ;;
19     *)     PARAMETER_FAULT=1 ;;
20     esac
21 fi
22
23 if [ $PARAMETER_FAULT -ne 0 ] ; then
24     $LOG " ${NAME}: chamado com :$*: ==> erro no parâmetro informado, abortando"
25     echo "`date +%Y%m%d%H%M%S` ${NAME}: erro no parâmetro informado, abortando"
26     exit 1
27 fi
28
29 ACTION_FAIL_START="/etc/init.d/heartbeat stop"
30 ACTION_OK_AGAIN="/etc/init.d/heartbeat start"
31
32 case $PARAMETER in
33     start) $LOG ${ACTION_FAIL_START} ;
34           ${ACTION_FAIL_START} ;;
35     stop) $LOG ${ACTION_OK_AGAIN} ;
36           ${ACTION_OK_AGAIN} ;;
37     esac
38 exit 0

```

Duas instalações do Linux-HA

Na **figura 1** vemos uma empresa qualquer, que chamaremos simplesmente de *cliente*, à esquerda. À direita temos um provedor de serviços terceirizado. Tanto o lado do cliente (esquerda, *gw-aa*) quanto o provedor de serviços (direita, *gw-ba*) usam a tecnologia ESP (*Encapsulating Security Payload* ou *área de dados encapsulada de forma segura*, um protocolo do IPsec)

para enviar pacotes pelo túnel. O túnel de reserva (Túnel 2) está configurado nos dois roteadores auxiliares no alto do diagrama (*gw-ab* e *gw-bb*) de uma maneira bastante semelhante à do Túnel 1.

O Linux-HA está em operação interligando *gw-aa* com *gw-ab* e também *gw-ba* com *gw-bb*. O *heartbeat* usa a porta serial, que não é usada para nenhum outro propósito e, por isso, é independente da rede, do IPsec e do IPTables. Ambas as

instalações de alta disponibilidade trabalham de forma independente uma da outra. Em condições normais os *gateways gw-aa* e *gw-ba* possuem endereços IP locais (10.1.255.254 à esquerda e 10.31.0.254 à direita). Caso haja alguma falha, esses endereços “migram” para os *gateways gw-ab* e *gw-bb*. O problema é que, se os IPs “migram” de um servidor a outro, os administradores perdem a capacidade de se conectar remotamente a eles, pois precisam de um IP fixo e imutável para isso. Para resolver o problema, um segundo endereço IP – esse sim, imutável – é atribuído à placa de rede da cada uma das máquinas envolvidas. Cada máquina possui, portanto, dois IPs: um “migratório” (ou seja, compartilhado) e o outro fixo e exclusivo. Quando o administrador quiser se conectar a um servidor em especial, basta usar o IP exclusivo. Em nosso exemplo, o endereço estático de *gw-bb* é 10.31.0.252.

Escrevendo scripts para uma VPN de alta disponibilidade

Os *gateways* padrão, *gw-aa* e *gw-ba*, rodam um shell script como o mostrado na **listagem 1**. O script é iniciado como um item no arquivo `/etc/inittab` e monitora a acessibilidade do outro lado, independentemente do túnel IPsec. Para isso, o script manda um pequeno pacote UDP para a porta echo (porta número 7). O protocolo *Echo* é um componente padrão em muitas distribuições Linux e pode usar UDP se assim configurado. Com isso, evitamos usar pacotes ICMP Echo Request – o “ping” normal – que são via de regra bloqueados por firewalls zelosos. Observe que, entretanto, a porta 7 também pode ser usada para realizar uma negação de serviço na máquina, portanto recomendamos cautela e monitoramento constante.

Assim que a conexão entre *gw-aa* e *gw-ba* esteja ativa, o contador `FAIL` será sempre zero. Se não houver resposta a

um dos nossos pings, o script vai incrementar, na **linha 56**, a variável `FAIL` em uma unidade – mas só se o valor estiver abaixo do configurado em `MAXFAIL`. Se o próximo ping estiver OK, o script reinicializa o valor de `FAIL` novamente a zero (**linha 65**). Se o valor atinge `MAXFAIL`, a **linha 53** chama o programa definido em `ACTION_FAIL_START` (no caso, o script mostrado na **listagem 2**, ao qual é passado o parâmetro `start`). O programa chamado por `ACTION_FAIL_START` para a “pulsção” local, forçando o gateway de reserva a adotar automaticamente o IP do roteador que “caiu”.

Esperando pela operação normal

O loop infinito se mantém rodando enquanto espera que o *gateway* principal volte à vida. Quando isso acontece, espera pelo “ping” da porta echo. Quando a primeira resposta chega, depois de uma falha de comunicação bastante longa, a conexão pode não ter, ainda, se estabilizado. Por isso, o script espera um bocadinho antes de recolocar todos os roteadores na condição normal de operação. Para esperar pelo período de histerese (espera), o script atribui um valor negativo a variável `FAIL` – mais precisamente, o valor presente na variável `HYSTERE` (**linha 62**) – e incrementa o valor de `FAIL` para cada ping que responda corretamente (**linha 70**). Se outro erro ocorrer na fase de recuperação, a **linha 46** coloca em `FAIL` um valor maior que o configurado em `MAXFAIL` – com isso, o sistema continua usando os roteadores de backup até que a conexão principal esteja normalizada e estável.

As coisas não voltam ao normal até que `FAIL` possua o valor zero. Só então o script chama `ACTION_OK_AGAIN` na **linha 75**. Esse programa também reinicia a pulsção (*heartbeat*) e reatribui os endereços IP corretos a todos os roteadores para funcionarem na condição inicial.

Essa abordagem multinível impede o “roteamento flipflop”, no qual os *gateways* ficam alternando entre os túneis cíclica e rapidamente. As conexões de Internet são propensas a falhas de poucos segundos. Se você procurar em seus próprios *logs* vai encontrar vários exemplos. Três minutos é um tempo de espera que consideramos ser bastante conveniente.

Nossa experiência em instalações desse tipo mostra que as falhas costumam sempre seguir um certo padrão. Se a pulsção em um lado do túnel cai, ela também será interrompida na outra ponta depois de um segundo – obviamente, os relógios dos dois sistemas têm que estar sincronizados para que consigamos perceber esses eventos olhando os registros (*logs*) do sistema.

Satisfação garantida

A solução descrita no artigo está funcionando desde o início de 2004 e demonstrou repetidamente que os usuários sequer notaram as falhas quando estas ocorreram. Em caso de um cataclisma, os mecanismos de reenvio de dados das pilhas TCP/IP dos servidores e clientes podem facilmente “aguentar” por volta de dez segundos – tempo suficiente para que o sistema de reserva consiga estabilizar-se e entrar em ação.

A conexão de reserva não usa a função de monitoração, o que poderia ser um problema. Na prática, se um provedor diferente for usado para a conexão de reserva, é muito difícil que ambos os provedores saiam do ar ao mesmo tempo. E, mesmo que isso aconteça, nem o protocolo BGP poderá ajudá-lo. ■

INFORMAÇÕES

- | |
|--|
| [1] Freeswan: www.freeswan.ca/code/super-freeswan/ |
| [2] RFC 1745: www.ietf.org/rfc/rfc1745.txt |
| [3] SSPE: sspe.sourceforge.net |
| [4] Linux-HA: www.linux-ha.org |
| [5] RFC 2401: www.ietf.org/rfc/rfc2401.txt |

Bancos de dados em C

Programando com GDBM

Deixe que o GNU Database Manager gerencie os dados de seus aplicativos e tenha menos dores de cabeça.

POR LUCAS BRASILINO

O GDBM, ou *GNU Database Manager* [1], é uma biblioteca escrita na linguagem C que possibilita ao programador a manipulação de informações em arquivos de banco de dados. Foi desenvolvido como uma alternativa GNU ao tradicional DBM, que fazia parte das primeiras versões do Unix. Existem outras bibliotecas similares à GDBM, como por exemplo a famosa *Berkeley DB* [2], mantida pela Sleepycat Software, a QDBM [3], mantida por Mikio Hirabayashi e a CDB [4], mantida por Daniel Bernstein, entre outras.

As principais vantagens do uso do GDBM são a facilidade, a velocidade e pouco consumo de recursos do sistema, graças ao uso de algoritmos *hash* [5]. Entre os softwares conceituados que o adotaram estão o OpenLDAP, o Sendmail e o Cyrus SASL.

Como já foi dito, o GDBM é voltado a programadores e não a usuários, ao contrário dos conhecidos MySQL, PostgreSQL e outros. Com ele é possível “embutir” re-

ursos de banco de dados em nossos programas. Linguagens como PHP, Python e Perl possuem funções para utilizá-lo.

Neste artigo abordaremos seu uso na linguagem C. Os programas `delete.c`, `fetch.c`, `seq.c`, `store.c` e `teste.c`, mencionados ao longo do texto, podem ser baixados no site da Linux Magazine, em [6]

Nas entranhas do GDBM

O armazenamento das informações é realizado na forma de chave/dado, em que a chave nunca pode ser repetida em um mesmo banco. Por exemplo, poderíamos criar um banco que contivesse as informações da **tabela 1**.

Seria impossível ter outra entrada com a chave “2222”. Se tentássemos inseri-la, o GDBM retornaria um código de erro. No máximo poderíamos substituí-la. Trataremos disso mais adiante.

Cada banco de dados é um arquivo chamado de *banco de dados GDBM*. É possível que vários processos simultâneos abram o mesmo arquivo no modo de leitura. Já no modo de gravação só é permitido o acesso a um processo por vez.

Tanto o dado quanto a chave devem ser do tipo *datum*, que é uma estrutura definida apenas com dois membros: `dptr` e `dsize`. O membro `dptr` é um apontador para caracter (`char *`) onde deve-se apontar para a chave ou o dado. O membro `dsize` é um inteiro (`int`) que

deve armazenar o comprimento da chave ou dado apontado por `dptr`. A **listagem 1** demonstra sua utilização.

Note que devemos sempre incluir o arquivo *header* (cabecalho) `gdbm.h` em nossos programas, pois nele estão declarados todos os protótipos das funções que iremos abordar, bem como os objetos, constantes (*flags*) etc.

A função `strlen()` retorna o comprimento da cadeia de caracteres. Para maiores informações sobre ela execute o comando `man 3 strlen` no Shell.

Em programação C os objetos são tecnicamente estruturas. Como gosto das nuances da programação orientada a objetos, durante este artigo iremos abordar estruturas como objetos.

Criando e abrindo um banco de dados

Para abrir um banco de dados, ou criá-lo caso não exista, utilizamos a função `gdbm_open()`. Essa função nos retorna um objeto do tipo `GDBM_FILE` que obrigatoriamente deve ser utilizado em todas as chamadas

Listagem 1: uso do tipo *datum*

```
01 char chave[] = "1111", dado[] = "simone";
02 datum key, data;
03 key.dptr = chave;
04 key.dsize = strlen(chave);
05 data.dptr = dado;
06 data.dsize = strlen(dado);
```

Tabela 1: exemplo de banco de dados

Matrícula (chave):	Nome (dado)
1111:	Simone
2222:	Cabral
3333:	Pedro
4444:	Augusto



por Nik Frey - www.sxc.hu

Listagem 2: criação de um banco de dados

```

01 #include <stdio.h>
02 #include <gdbm.h>
03 int
04 main (void)
05 {
06     GDBM_FILE db;
07     db = gdbm_open ("database.db", 512, GDBM_WRCREAT, 0644, NULL);
08     if (db == NULL)
09     {
10         fprintf (stderr, "Não pude abrir/criar banco de dados!\n");
11         return 1;
12     }
13     gdbm_close (db);
14     return 0;
15 }

```

subseqüentes a essa biblioteca, excluindo-se apenas a chamada `gdbm_strerror()`, que retorna uma frase identificando um erro ocorrido. Veja na **listagem 2** como criar um novo banco.

Certamente esse programa não é muito útil, pois estamos abrindo (ou criando) um banco de dados, testando se ele foi mesmo aberto e imediatamente fechando-o. Seu intuito é ser puramente didático.

A função `gdbm_open()` recebe 5 parâmetros. São eles:

- ⇒ O nome do arquivo, podendo-se incluir o caminho completo.
- ⇒ Tamanho do bloco em bytes. A cada acesso ao arquivo, o GDBM transfere os dados em blocos de tamanho definido neste parâmetro. 512 bytes é o menor tamanho.
- ⇒ Uma *flag* que determina a forma da abertura do arquivo.
- ⇒ O modo (permissões de acesso), em notação octal.
- ⇒ Uma função, declarada na forma `void func()`, que o GDBM chamará caso encontre um erro fatal. `NULL` fará com que seja chamada uma função padrão.

Existem várias flags disponíveis, são elas:

- ⇒ `GDBM_WRITER`: o programa poderá tanto ler quanto escrever no arquivo, porém deverá ter acesso exclusivo.
- ⇒ `GDBM_WRCREAT`: semelhante ao `GDBM_WRITER` mas, se o arquivo não existir, será criado.
- ⇒ `GDBM_READER`: o programa poderá apenas ler dados do arquivo. É permitida a leitura por múltiplos programas simultaneamente.
- ⇒ `GDBM_NEWDB`: semelhante à `GDBM_WRCREAT`, porém sempre criará um arquivo novo, sobrescrevendo um antigo com o mesmo nome, caso exista. Use-a com cuidado.

Essas flags podem ser combinadas através de um operador lógico `OR (|)` com duas outras flags bem específicas:

- ⇒ `GDBM_SYNC`: Sincroniza todas as operações com o arquivo. O ponto positivo é que evitaremos corrupção de dados caso o sistema seja desligado indevidamente e o ponto negativo é que perdemos, e muito, em desempenho.
- ⇒ `GDBM_NOLOCK`: Desliga as rotinas de *locking* gerenciadas pela biblioteca. O próprio programa deverá fazê-lo.

Por exemplo, poderíamos utilizar `gdbm_open()` para abrir o banco `database.db` para escrita no modo de sincronismo:

```
db = gdbm_open ("database.db", 512, GDBM_WRITER | GDBM_SYNC, 0644, NULL);
```

Por padrão o GDBM abre um arquivo no modo assíncrono e gerencia seu próprio *locking*.

Na **listagem 2** vemos que a chamada ao `gdbm_open()` retorna `NULL` caso haja algum erro. Os erros mais comuns são a falta de permissão de acesso ao arquivo e a tentativa de acesso nos modos `GDBM_WRITER` ou `GDBM_WRCREAT` com um arquivo já aberto por outro processo.

A função `gdbm_close()`, como o nome já sugere, fecha um arquivo de banco de dados. Ela deve ser sempre chamada antes do fim do programa. Assim, o GDBM sincroniza todos os dados entre memória e o arquivo em disco evitando qualquer corrupção dos dados e libera áreas de memória alocadas para o objeto `GDBM_FILE`.

Armazenando um dado

O programa `store.c` contém uma rotina completa de exemplo de como um par chave/dado é armazenado utilizando o GDBM. Para compilá-lo, devemos executar o comando:

```
$ gcc store.c -o store -lgdbm
```

A opção `-lgdbm` diz ao *link-editor* para ligar o binário à biblioteca GDBM (*libgdbm*). Em seguida, basta executá-lo passando como argumentos do binário `store` a chave e o dado:

```
$ ./store 1111 simone
```

Ao se listar o conteúdo do diretório corrente, o arquivo `database.db` deve ser encontrado o arquivo. Ele é o nosso banco de dados. Quem realmente armazena o par chave/dado é a função `gdbm_store()`:

```
ret = gdbm_store (db, key, data, GDBM_INSERT);
```

Essa função recebe 4 parâmetros:

- ⇒ o objeto relacionado a um arquivo (*db*);
- ⇒ a chave (*key*), previamente preenchida;
- ⇒ o dado (*data*), previamente preenchido;
- ⇒ e uma flag.

Essa flag pode assumir dois valores:

- ⇒ `GDBM_INSERT`: insere um par chave/dado novo, retorna erro caso a chave exista;
- ⇒ `GDBM_REPLACE`: insere ou substitui um par chave/dado.

A função `gdbm_store()` retorna um valor inteiro indicando se sua execução foi realizada ou não a contento. Caso retorne `0`, o par chave/dado foi inserido/substituído. Caso retorne `1`, o par chave/dado não foi inserido, provavelmente porque foi utilizada a flag `GDBM_INSERT` e a chave já existia no arquivo. Caso retorne `-1`, o

armazenamento não foi possível porque o banco de dados foi aberto apenas para leitura (flag `GDBM_READER`) ou o objeto do tipo `GDBM_FILE` é `NULL` (o arquivo não foi aberto/criado) ou o membro `dptr` da chave ou dado é `NULL`.

Como exercício, insira todas as chaves/dados descritos na **tabela 1**.

Buscando um dado

O programa `fetch.c` nos mostra como podemos buscar um dado referente a uma chave no arquivo. Basta chamar a função `gdbm_fetch()`:

```
data = gdbm_fetch (db, key);
```

Ou seja, ao passarmos a chave nos é retornado o dado, lembrando que este se encontra no membro `dptr`. *Um ponto muito importante:* o GDBM usa a função `malloc()` para alocar dinamicamente

uma área de memória para armazenar o dado, portanto devemos liberá-la com a função `free()`, como podemos ver no código-fonte do `fetch.c`:

```
free (data.dptr);
```

Compile e execute-o, passando a chave como argumento:

```
$ gcc fetch.C -o fetch -lgdbm
$ ./fetch 1111
Chave: 1111
Dado: simone
```

Excluindo um dado

Excluir um par chave/dado também é simples, basta utilizar a função `gdbm_delete()` informando uma chave. Para exemplificar, vamos utilizar o programa `delete.c`. Compile-o de forma semelhante aos programas mostrados anteriormente. ➡

Como pode ser visto no código fonte, a exclusão está sendo feita pela chamada:

```
ret = gdbm_delete (db, key);
```

Essa chamada também retorna um valor inteiro, 0 caso a exclusão tenha sido realizada e -1 caso não tenha sido realizada: nesse caso, ou a chave não foi encontrada ou o objeto `db` não é válido.

Vamos adicionar um dado e imediatamente excluí-lo:

```
$ ./store 5555 vilaca
$ ./delete 5555
```

Acessando todo o banco de dados

O GDBM provê duas funções interessantes que nos permitem acessar todos os pares chave/dado. São as funções `gdbm_firstkey()` e `gdbm_nextkey()` que retornam, respectivamente, a primeira chave na estrutura interna do GDBM e a próxima chave.

É necessário esclarecer que a primeira chave a ser recuperada não será necessariamente a primeira chave que armazenamos, nem que as chaves serão recuperadas na ordem de armazenagem.

A idéia é simples: de posse da primeira chave retornada pela função `gdbm_firstkey()` faz-se um laço chamando repetitivamente `gdbm_nextkey()` até este retornar `NULL`, indicando que não há mais chaves a retornar. Essa rotina seria a seguinte:

```
for (key = gdbm_firstkey (db); key.dptr;
     key = gdbm_nextkey (db, key))
{
    /* algum código */
};
```

`gdbm_firstkey()` apenas recebe um argumento, já `gdbm_nextkey()` recebe dois. Seu segundo argumento é a chave acessada pela chamada anterior.

O programa `seq.c` utiliza esta rotina. Compile-o e rode-o como já explicado:

```
$ ./seq
Chave: 2222
Dado: cabral
Chave: 3333
Dado: pedro
Chave: 1111
Dado: simone
Chave: 4444
Dado: agosto
```

Utilizando essas duas funções, o GDBM nos garante que todos os pares chave/dados serão acessados.

Erros

O GDBM possui seu próprio conjunto de mensagens de erro que podem ser acessadas pela função `gdbm_strerror()`. Em todos os exemplos de programas neste artigo esse conjunto foi utilizado para indicar algum problema. O único parâmetro dessa função é a variável inteira global `gdbm_errno` que é definida no arquivo `gdbm.h` (por isso não tivemos que declará-la).

O uso de `gdbm_strerror()` não é obrigatório, mas é bastante cômodo.

Funções adicionais

Existem algumas funções que raramente são usadas, mas é interessante conhecê-las. A necessidade de sua utilização depende diretamente da forma como o programa foi escrito, e não da biblioteca em si.

A primeira é `gdbm_sync()`, que tem como argumento o objeto do banco de dados (`GDBM_FILE`). Como por padrão um arquivo é aberto na forma assíncrona, essa função força a sincronização entre os dados que estão num *buffer* na memória e o arquivo no disco. Ela é automaticamente chamada quando fechamos o banco (`gdbm_close()`). Portanto só é necessário usá-la se muitas operações de inclusão e exclusão foram feitas sem fechar e reabrir o arquivo.

A segunda é `gdbm_reorganize()`, que recebe o mesmo argumento que a função anterior. Sua utilidade é diminuir o tamanho do arquivo quando se fizeram muitas exclusões. Na realidade o GDBM não exclui “fisicamente” um par chave/dado do arquivo quando utilizamos `gdbm_delete()`; ele apenas marca aquele par como deletado e nunca mais o acessa, porém os dados continuam a ocupar espaço.

A terceira é `gdbm_setopt()`, que é utilizada para ajustar algumas opções da biblioteca, como ligar ou desligar o modo síncrono, tamanho do buffer interno etc.

A última é `gdbm_fdesc()`, utilizada para permitir ao programador avançado gerenciar ele próprio o “locking” de seus arquivos, desde que os mesmos tenham sido abertos com a flag `GDBM_NOLOCK`.

O GDBM esconde atrás de sua interface simples um mecanismo rápido e robusto de armazenamento e recuperação de dados. Suas possibilidades só são limitadas pela imaginação do programador. Ele também pode ser instrutivo para quem deseja se aventurar com bibliotecas similares, como a Berkeley DB. Mãos à obra! ■

SOBRE O AUTOR

Lucas Brasilino começou a usar Linux nos idos de 95 e não largou mais. Trabalhou na LBS como instrutor e consultor e na Emprtel (Prefeitura do Recife) na área de suporte a redes e servidores rodando Software Livre. É desenvolvedor do SIRI – Sistema Integrado de Gestão de Recursos de Internet. [7]



INFORMAÇÕES

[1] www.gnu.org/software/gdbm

[2] www.sleepycat.com/products/db.shtml

[3] qdbm.sourceforge.net

[4] cr.yip.to/cdb.html

[5] en.wikipedia.org/wiki/Hash_function

[5] sl.recife.pe.gov.br/projects/siri

[6] www.linuxmagazine.com.br/Magazine/current

linuxUSER

BEM-VINDO À LINUX USER!



Esta é uma seção especial dedicada a destacar programas úteis e interessantes para ajudá-lo no seu trabalho diário com o Linux no desktop. Aqui você encontrará informações sobre como utilizar programas comuns de forma mais eficiente, obterá um valioso embasamento técnico e conhecerá as últimas novidades em software para seu sistema operacional favorito.

Truques rápidos 80

Softwares são como pessoas: há detalhes e segredinhos que você só descobre depois de anos de relacionamento. Conheça aqui algumas dicas para softwares populares no mundo Linux que podem facilitar sua vida e poupar tempo.

Papo de Botequim 84

Vamos aprender mais sobre formatação de cadeias de caracteres, conhecer as principais variáveis do Shell e nos aventurar no (ainda) desconhecido território da expansão de parâmetros.

Os segredinhos do pingüim

Truques rápidos

Softwares são como pessoas: há detalhes e segredinhos que você só descobre depois de anos de relacionamento. Conheça aqui algumas dicas para softwares populares no mundo Linux que podem facilitar sua vida e poupar tempo.

POR ANDREA MÜLLER E THOMAS HÜMMLER

★ Trabalhando com arquivos do Microsoft Office

O *OpenOffice.org* abre, processa e salva textos, planilhas e apresentações do *Microsoft Office*. Arquivos do *Word* (extensão *.doc*), do *Excel* (*.xls*) ou do *Powerpoint* (*.ppt* ou *.pps*) podem ser abertos a partir do menu *Arquivo* | *Abrir...*; Na janela que surge, basta selecionar o arquivo e clicar no botão *Abrir*.

Para salvar um único arquivo no formato Microsoft Office selecione *Arquivo* | *Salvar como...* e procure o formato adequado na lista *Salvar como tipo*. Se você quiser salvar os textos, planilhas e apresentações sempre no formato correspondente do MS Office há mais uma opção: selecione em *Ferramentas* | *Opções...* o item *Carregar/Salvar* | *Geral*. Na área *Formato de arquivo padrão* escolha o tipo de documento no campo *Tipo de documento*. No menu do lado direito (*Sempre salvar como*)

selecione o formato desejado para aquele tipo de documento (ver [figura 1](#)). Observação: o *OpenOffice.org* não suporta funções especiais – como por exemplo "lembrar" a posição do cursor – quando o documento é salvo em outros formatos que não o nativo.

A partir desse momento, ao salvar um documento, o tipo de arquivo será ajustado de acordo com essa seleção. Naturalmente, outro tipo de arquivo pode ser selecionado manualmente no menu *Arquivo* | *Salvar como...* ■

★ Convertendo documentos para OpenOffice.org

Se você tiver muitos arquivos antigos como textos no formato do *Word*, planilhas no formato do *Excel* ou apresentações no formato do *Powerpoint*, pode querer transformá-los para o formato *OpenOffice.org* correspondente. Arquivos antigos no formato *StarOffice* também podem ser convertidos do mesmo modo. Para isso existem duas possibilidades: abrir cada documento separadamente e salvar no formato *OpenOffice.org* ou automatizar esse procedimento com a ajuda do *Conversor de documentos*. Com ele, todos os arquivos do *StarOffice* ou do *Microsoft Office* de um diretório são rapidamente convertidos para formatos *OpenOffice.org*.

Para iniciar o assistente adequado selecione *Arquivo* | *Assistente* | *Conversor de documentos...* Primeiramente marque os tipos de documentos que serão modificados. Por padrão, estará marcada a opção *StarOffice*; se os documentos que serão convertidos forem estiverem nos formatos *Word*, *Excel* ou *PowerPoint* clique em *Microsoft Office* e selecione os tipos de documento desejados.

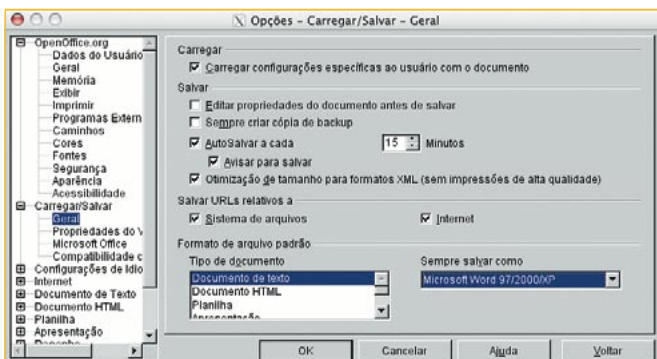


Figura 1: Determine o formato padrão em que o *OpenOffice.org* salvará os documentos em *Ferramentas* | *Opções* | *Carregar/Salvar* | *Geral*.

A opção *Criar arquivo de log* forma um registro da conversão. Deixe esta opção ativada para reconhecer eventuais erros. Clique em *Próximo* para passar à próxima tela.

Nessa tela, você deve informar se a conversão será de modelos, documentos ou ambos. No campo *Importar de...* selecione o diretório onde estão os documentos que serão convertidos. No campo *Salvar em...* especifique para onde os arquivos serão transferidos. Se pretende converter apenas os arquivos de um diretório, desative a opção *Incluindo subdiretórios*.

Clique em *Próximo*. Ajuste então, para o próximo tipo de documento escolhido, quais arquivos de quais formatos serão salvos em quais formatos finais. Quando as definições para todos os tipos de documentos tiverem sido feitas, aparece um resumo. Nessa tela será exibido aquilo que o assistente fará quando for iniciado (com um clique no botão *Converter*). Na seqüência, ele captura todos os documentos, converte-os e registra todos os procedimentos no arquivo de log, se essa opção não estiver desativada.

Esses procedimentos podem ser verificados na opção *Mostrar arquivo de log* após a conversão de todos os documentos. Dispense então o assistente para que ele volte a seus aposentos, clicando em *Fechar*. ■

★ Manter ou apagar as macros do Microsoft Office?

O Microsoft Office e o OpenOffice.org trabalham com diferentes linguagens de macro. O Microsoft Office utiliza *Visual Basic for Applications* (VBA) e o OpenOffice.org faz uso do *StarBasic* como interface de programação. Ambos são baseados na mesma linguagem, o velho e bom BASIC (*Beginner's All-Purpose Symbolic Instruction Code* ou código de instruções simbólicas de uso geral para principiantes), porém existem diferenças entre métodos e objetos. Por esse motivo as macros do Word, do Excel ou do Powerpoint não podem ser usadas no Writer, no Calc ou no Impress, respectivamente.

As macros existentes do Visual Basic serão transcritas automaticamente no *Editor Basic* do OpenOffice.org e podem ser utilizadas novamente, se o usuário conhecer as duas linguagens o suficiente. Nesse caso, os ajustes iniciais devem ser mantidos, pois o OpenOffice.org carrega as macros automaticamente. Porém, todas as linhas estão precedidas por marcas de comentários – e estão, portanto, inativas.

Normalmente é mais rápido, pelo menos em macros simples, gravá-las novamente direto no OpenOffice.org. Para isso há dois ajustes através de *Ferramentas | Opções* no item *Carregar | Salvar | Propriedades do VBA* (ver **figura 2**). Aqui você determina o que deve acontecer com as macros gravadas em documentos de Word, Excel e PowerPoint.

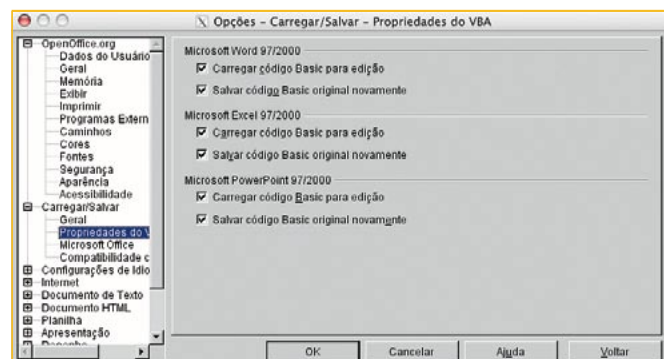


Figura 2: O comportamento do código de macros do Microsoft Office pode ser controlado através de *Ferramentas | Opções | Carregar/Salvar | Propriedades do VBA*.

⇒ *Carregar código Basic para edição:* Ao importar um documento no formato da Microsoft, o OpenOffice.org carrega o código da macro e o salva como módulo Basic do OpenOffice.org. Este código pode então ser editado no *Editor Basic*. O documento original deve ser salvo em formato OpenOffice.org, para que o programa salve o código da macro; em outros formatos, ele será deixado de fora.

⇒ *Salvar o código Basic original novamente:* Ao salvar um documento em formato Microsoft, as macros são salvas sem modificação e o Microsoft Office poderá executá-las posteriormente. Em outros formatos, como no formato OpenOffice.org, a macro não será salva. Se os dois campos estiverem como no ajuste inicial e o código de macro com marcas de comentários for editado através do OpenOffice.org com o *Editor Basic*, o código Visual Basic original só será preservado ao salvar o documento no formato Microsoft. Se a opção *Salvar o código Basic original novamente* for desativada, eventuais vírus de macro, comuns no mundo Windows®, existentes nos documentos serão facilmente apagados. ■

★ Qual é o tipo de arquivo?

Mesmo com a Microsoft clamando aos quatro ventos que seu conjunto de aplicativos de escritório usa o formato de arquivo XML desde 2003, para o OpenOffice.org isto já é coisa antiga. Desde a primeira versão o OoO utiliza este formato para salvar seus documentos. Os arquivos XML do OpenOffice diferenciam-se pela extensão do arquivo:

- ⇒ *sxw*: Documentos do Writer
- ⇒ *stw*: Modelos do Writer
- ⇒ *sxc*: Planilhas do Calc
- ⇒ *stc*: Modelos do Calc
- ⇒ *sxi*: Apresentações do Impress
- ⇒ *sti*: Modelos do Impress
- ⇒ *sxd*: Desenhos do Draw

- ⇒ *std*: Modelos do Draw
- ⇒ *sxm*: Fórmulas do Math
- ⇒ *sxg*: Documento mestre

Assim, se você quiser, por exemplo, localizar todos os desenhos do Draw, procure pela extensão de arquivo correspondente (*.sxd*). Se quiser localizar todos os modelos do OpenOffice.org no *KFind* (a ferramenta de busca de arquivos do KDE) procure por **.st?*. ■

★ Protetor de tela engatilhado

Da mesma forma que nos sistemas *Windows*®, os protetores de tela do KDE são arquivos executáveis. Porém, eles não levam a extensão *.scr* mas sim *.kss* (de **KDE Screen Saver**). Pode-se encontrar todos os módulos de protetores de tela disponíveis se procurarmos a partir do diretório raiz por todos os arquivos com esta extensão (**.kss*) com o *KFind* (*Menu K | Procurar arquivos*).

Você pode iniciar um protetor de tela diretamente no modo de teste chamando-o pela janela *Executar Comando...* do Menu K (que também pode ser acessada com a combinação de teclas **[Alt] + [F2]**). Basta digitar o nome do arquivo e pressionar **[Enter]** – por exemplo, *keuphoria.kss* (veja a **figura 3**). Se você adicionar o parâmetro *--setup* ao comando é possível acessar a caixa de diálogo de configuração do protetor de tela e modificar seu comportamento. ■

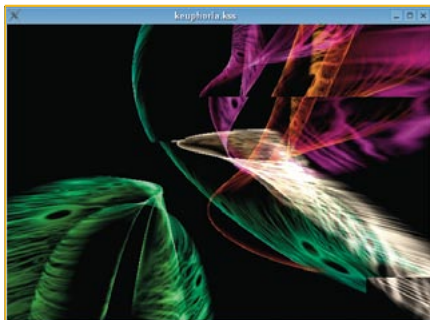


Figura 3: Quando se inicia um protetor de tela de uma janela de início rápido, ele rodará em modo de demonstração em uma janela própria..

★ Busca relâmpago com o Kfind

O *KFind* é o utilitário do KDE para localização de arquivos no seu computador. Você pode chamá-lo pelo menu K (opção *Procurar arquivos*) ou, na linha de comando, digitando *kfind*.

À medida que o seu computador vai sendo usado, a quantidade de arquivos no disco rígido aumenta progressivamente. Para que a busca por um arquivo não leve tempo demais, habilite na aba *Nome/Localização* a opção *Usar índice de arquivos*. Dessa forma, o *KFind* utiliza como referência um arquivo de índice, no qual ele encontrará o arquivo desejado num piscar de olhos.

Para isso, o programa usa a ferramenta *locate*, presente em praticamente todas as distribuições Linux, exceto o SUSE. Caso você seja fã do camaleão, vá ao *menu K | Sistema* e selecione o item *YaST*. Na janela do YaST selecione o módulo *Instalar e Remover Software*. Na tela que surge escreva *locate* no campo *Pesquisa*: e clique no botão *Pesquisa*. Na lista de resultados, selecione o pacote *findutils-locate* (basta clicar no quadradinho em frente ao nome) e clique no botão *Aceitar*. Você vai precisar dos CDs do SUSE LINUX à mão (ou uma conexão à Internet).

Entretanto, nada é perfeito. Na prática, a procura no índice possui um pequeno porém: seu sistema Linux atualiza o arquivo de índice, também conhecido como *banco de dados locate* (*locate database*, ou *locatedb*), somente uma vez por dia. Essa atualização é feita como uma tarefa agendada pelo subsistema *cron*, que o Linux inicia automaticamente em segundo plano, geralmente durante a madrugada. Portanto, somente ficam no banco de dados do *locate* os arquivos inseridos antes da última atualização. Se você quiser forçar a atualização do índice, faça isso com o comando *updatedb*, que pode ser digitado em um terminal. O comando precisa de um certo tempo

para ser executado e faz muito acesso ao disco rígido, portanto não se surpreenda se ele der umas "estaladas". ■

★ Restaurando as configurações do Mozilla e Firefox

Se você modificou um dos ajustes ocul-tos do *Mozilla* ou *Firefox* através da tela *about:config*, não é necessário lembrar de todas as modificações para revertê-las ao valor inicial. As opções modificadas pelo usuário são mostradas em negrito, de modo que é fácil reconhecer onde você usou valores próprios. Para trabalhar novamente com os valores iniciais do navegador, clique com o botão direito do mouse sobre o registro modificado e selecione *Reset* no menu de contexto. ■

★ Pesquisando na Web

PCI, ACPI, USB, Emacs, NTFS, /etc: Em relação a computadores, especialmente a Linux, existem inúmeras expressões que o usuário, especialmente o iniciante, em muitos casos não consegue relacionar. Também em outras áreas, às vezes é necessário alguma intuição para encontrar a direção certa: “Blake foi apenas poeta ou será que também desenhava?”, “O que é mesmo um Hai-kai: uma forma de poesia japonesa ou um prato à base de peixe?”

O mecanismo de busca *Google* (www.google.com.br) oferece uma função muito prática e pouco conhecida para encontrar definições de expressões de forma rápida. Escreva no campo de pesquisa *define: expressão* para o Google procurar a explicação correta. Em muitos casos a página com os resultados já revela o mais relevante, sem que haja necessidade de abrir cada um deles. Um pequeno porém: a pesquisa de definições existe atualmente apenas para expressões em inglês, e os resultados são também dados em inglês. Uma alternativa é a excelente enciclopédia online *Wikipedia*, que tem versão em português (pt.wikipedia.org). ■

★ Mudar a cor das citações no Evolution

O *Evolution* mostra o texto citado em e-mails num tom cinza escuro, que praticamente não se destaca do texto normal. Se você preferir que as citações venham numa cor mais acentuada, não tem problema: abra a caixa de configuração em *Ferramentas | Configurações* e selecione o item *Preferências de correio*. Ative a opção *Realçar citações com cor* e escolha a cor desejada (essa opção está sob *Exibição de mensagens*). Abre-se então uma caixa com um círculo de cores (ver [figura 4](#)) no qual você pode escolher o tom desejado para as citações.

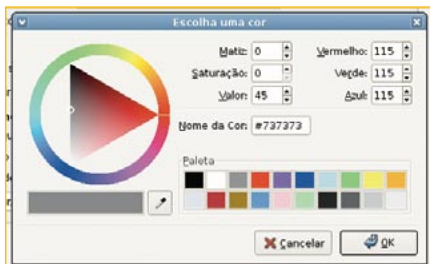


Figura 4: Através do campo de escolha de cores você define como o Evolution mostrará as citações..

A cor desejada pode ser selecionada ainda mais facilmente com o uso do "conta-gotas". Clique uma vez nesse ícone. O ponteiro do mouse agora tem forma de um conta-gotas. Quando você clica em qualquer ponto da área de trabalho, o conta-gotas "chupa" a cor daquele ponto e a define como cor das citações. Feche a caixa de seleção e depois a de configurações clicando em *OK*. Daí em diante, o Evolution mostrará o texto citado na cor recém-selecionada. ■

★ Manpages: Classificação por temas

A documentação clássica no Linux (e em qualquer Unix) é organizada e classificada nas famosas *manpages* (páginas de manual). Em modo texto, elas são exibidas com o comando `man nome_do_programa`.

Com *[seta para cima]* e *[seta para baixo]* percorre-se o texto; Use a tecla *[q]* para fechar o documento.

As páginas de manual não são armazenadas todas em um único índice, mas numa pasta sob `/usr/share/man`. Nela existem os índices *man1*, *man2* etc. Nesse caso, o número simboliza a seção na qual as páginas de manual se encontram. Para usuários normais, são interessantes as seções 1, 5 e 8, já que ali estão as páginas para programas de usuários, arquivos de configuração e comandos para administração do sistema., respectivamente.

Interessante também é ver as categorias individuais no *Centro de Ajuda do KDE*. Inicie o Centro de Ajuda com o comando `khelppcenter` (ou escolhendo a opção apropriada no menu K) e, na árvore à direita, clique em *unix-manpages*. Dessa forma, o programa mostra todas as seções com uma breve explicação sobre cada uma ([figura 5](#)). ■

★ Referências nas manpages

Além dos parâmetros opcionais e explicações de uso do programa, você também vai encontrar no final de muitas páginas de manual o item *SEE ALSO* (ou *VEJA TAMBÉM*, como mostrado na [figura 6](#)), que faz referência a outras partes da documentação do sistema. Além de indicações sobre como encontrar mais informação na Internet, há itens do tipo *nome_do_programa(x)*, sendo *x* sempre um número entre 1 e 8.

Por exemplo, a manpage do comando `ls` ([figura 3](#)) contém, entre outros dados, a referência *dircolors(1)*. Sempre que algo se apresentar nesse formato, seguramente é uma página de manual. Depois do nome (aqui, *dircolors*) temos, entre parênteses, a seção na qual a página se encontra (seção 1, no caso). Geralmente,

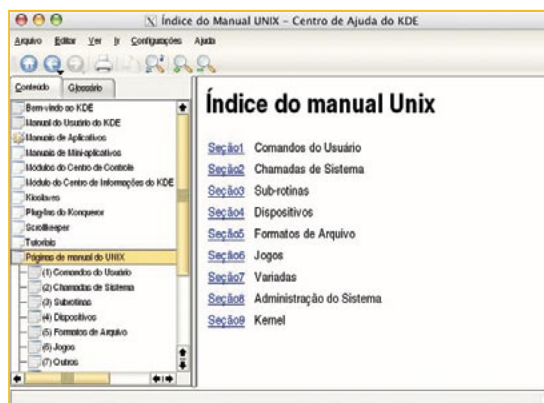


Figura 5: O programa `khelppcenter` exibe as seções usadas para classificar as páginas de manual.

basta chamar a página com o comando `man display`, não é preciso indicar a seção à qual ela pertence.

Só é necessário escrever explicitamente o número da seção quando há várias manpages com o mesmo nome – por exemplo, há uma página na seção 1 sobre o comando `if` e outra na seção 3 sobre a instrução `if`, da linguagem de programação Perl. Nesses casos, a chamada à página do comando seria: `man 1 if`. Sem essa informação, o comando `man` inicia sua busca na seção 1 e varre todas até a 8, exibindo a primeira manpage que encontrar. Caso existam nas seções muitas páginas com o mesmo nome, será possível que a página errada seja escolhida – por isso, indicar a seção apropriada pode poupar muita dor de cabeça ou tempo de pesquisa. ■



Figura 6: No item *VEJA TAMBÉM* de uma página de manual você encontrará referências a documentações complementares.



Curso de Shell Script

Papo de Botequim

Parte IX

Hoje vamos aprender mais sobre formatação de cadeias de caracteres, conhecer as principais variáveis do Shell e nos aventurar no (ainda) desconhecido território da expansão de parâmetros. E dá-lhe chope!

POR JÚLIO CEZAR NEVES

Tá bom, já sei que você vai querer chope antes de começar, mas tô tão a fim de te mostrar o que fiz que já vou pedir a rodada enquanto isso. Aê Chico, manda dois! O dele é sem colarinho pra não deixar cheiro ruim nesse bigodão...

Enquanto o chope não chega, deixa eu te lembrar o que você me pediu na edição passada: era para refazer o programa *listartista* com a tela formatada e execução em loop, de forma que ele só termine quando receber um [ENTER] sozinho como nome do artista. Eventuais mensagens de erro e perguntas feitas ao usuário deveriam ser mostradas na antepenúltima linha da tela, utilizando para isso as rotinas externas *mandamsg.func* e *pergunta.func* que desenvolvemos durante nosso papo na edição passada.

Primeiramente eu dei uma encolhida nas rotinas *mandamsg.func* e *pergunta.func*, que ficaram como na **listagem 1**. E na **listagem 2** você tem o “grandão”, nossa versão refeita do *listaartista*.

– Puxa, você chegou com a corda toda! Gostei da forma como você resolveu o problema e estruturou o programa. Foi mais trabalhoso, mas a apresentação ficou muito legal e você explorou bastante as opções do comando *tput*. Vamos testar o resultado com um álbum do *Emerson, Lake & Palmer* que tenho cadastrado.

Envenenando a escrita

Ufa! Agora você já sabe tudo sobre leitura de dados, mas quanto à escrita ainda está apenas engatinhando. Já sei que você vai me perguntar: “Ora, não é com o comando *echo* e com os redirecionamentos de saída que se escreve dados?”.

Bom, a resposta é “mais ou menos”. Com estes comandos você escreve 90% do que precisa, porém se precisar escrever algo formatado eles lhe darão muito trabalho. Para formatar a saída veremos agora uma instrução muito mais interessante, a *printf*. Sua sintaxe é a seguinte:

Listagem 1: *mandamsg.func* e *pergunta.func*

mandamsg.func

```
01 # A função recebe somente um parâmetro
02 # com a mensagem que se deseja exibir.
03 # Para não obrigar o programador a passar
04 # a msg entre aspas, usaremos $* (todos
05 # os parâmetro, lembra?) e não $1.
06 Msg="$*"
07 TamMsg=${#Msg}
08 Col=$((TotCols - TamMsg / 2)) # Centra msg na linha
09 tput cup $LinhaMsg $Col
10 read -n1 -p "$Msg "
```

pergunta.func

```
01 # A função recebe 3 parâmetros na seguinte ordem:
02 # $1 - Mensagem a ser mostrada na tela
03 # $2 - Valor a ser aceito com resposta padrão
04 # $3 - O outro valor aceito
05 # Supondo que $1=Aceita?, $2=s e $3=n, a linha
06 # abaixo colocaria em Msg o valor "Aceita? (S/n)"
07 Msg="$1 (`echo $2 | tr a-z A-Z`/`echo $3 | tr A-Z a-z`)"
08 TamMsg=${#Msg}
09 Col=$((TotCols - TamMsg / 2)) # Centraliza msg na linha
10 tput cup $LinhaMsg $Col
11 read -n1 -p "$Msg " SN
12 [ ! $SN ] && SN=$2 # Se vazia coloca default em SN
13 SN=$(echo $SN | tr A-Z a-z) # A saída de SN será em minúscula
14 tput cup $LinhaMsg $Col; tput el # Apaga msg da tela
```

```
printf formato [argumento...]
```

Onde *formato* é uma cadeia de caracteres que contém três tipos de objeto: caracteres simples, caracteres para especificação de formato (ou *de controle*) e seqüência de escape no padrão da linguagem C. *argumento* é a cadeia de caracteres a ser impressa sob o controle de *formato*.

Cada um dos caracteres utilizados é precedido pelo caractere % e, logo a seguir, vem a especificação de formato de acordo com a **tabela 1**.

As seqüências de escape padrão da linguagem C são sempre precedidas pelo caractere contra-barras (\). As reconhecidas pelo comando printf são as da **tabela 2**.

Não acaba por aí não! Tem muito mais coisa sobre essa instrução, mas como esse é um assunto muito cheio de detalhes e, portanto, chato para explicar e pior ainda para ler ou estudar, vamos passar direto aos exemplos com comentários. Veja só:

```
$ printf "%c" "1 caractere"
1$
```

Errado! Só listou 1 caractere e não saltou linha ao final

```
$ printf "%c\n" "1 caractere"
1
```

Saltou linha mas ainda não listou a cadeia inteira

```
$ printf "%c caractere\n" 1
1 caractere
```

Listagem 2: listaartista

```
$ cat listaartista3.sh
01 #!/bin/bash
02 # Dado um artista, mostra as suas musicas
03 # versao 3
04 LinhaMesg=$((`tput lines` - 3)) # Linha onde as msgs serão mostradas
05 TotCols=$(tput cols)          # Qtd de colunas na tela para enquadrar msgs
06 clear
07 echo "
                                +-----+
                                | Lista Todas as Músicas de um Determinado Artista |
                                | ===== |
                                | Informe o Artista: |
                                +-----+"
08 while true
09 do
10     tput cup 5 51; tput ech 31 # ech=Erase chars (31 para não apagar barra vertical)
11     read Nome
12     if [ ! "$Nome" ]          # $Nome esta vazio?
13     then
14         . pergunta.func "Deseja Sair?" s n
15         [ $SN = n ] && continue
16         break
17     fi
18     fgrep -iq "^$Nome~" musicas || # fgrep não interpreta ^ como expressão regular
19     {
20         . mandamsg.func "Não existe música desse artista"
21         continue
22     }
23     tput cup 7 29; echo '|
24     LinAtual=8
25     IFS="
26 :"'
27     for ArtMus in $(cut -f2 -d^ musicas) # Exclui nome do album
28     do
29         if echo "$ArtMus" | grep -iq "^$Nome~"
30         then
31             tput cup $LinAtual 29
32             echo -n '| '
33             echo $ArtMus | cut -f2 -d~
34             tput cup $LinAtual 82
35             echo '| '
36             let LinAtual++
37             if [ $LinAtual -eq $LinhaMesg ]
38             then
39                 . mandamsg.func "Tecle Algo para Continuar..."
40                 tput cup 7 0; tput ed # Apaga a tela a partir da linha 7
41                 tput cup 7 29; echo '|
42                 LinAtual=8
43             fi
44         fi
45     done
46     tput cup $LinAtual 29; echo '|
47     tput cup=$((++LinAtual)) 29
48     read -n1 -p "+-----Tecle Algo para Nova Consulta-----+"
49     tput cup 7 0; tput ed          # Apaga a tela a partir da linha 7
50 done
```

Tabela 1: Formatos de caractere

Caractere	A expressão será impressa como:
c	Caractere simples
d	Número no sistema decimal
e	Notação científica exponencial
f	Número com ponto decimal (<i>float</i>)
g	O menor entre os formatos %e e %f com omissão dos zeros não significativos
o	Número no sistema octal
s	Cadeia de caracteres
x	Número no sistema hexadecimal
%	Imprime um %. Não há nenhum tipo de conversão

Opa, essa é a forma correta! O %c recebeu o valor 1, como queríamos:

```
$ a=2
$ printf "%c caracteres\n" $a
2 caracteres
```

O %c recebeu o valor da variável \$a.

```
$ printf "%10c caracteres\n" $a
          2 caracteres
$ printf "%10c\n" $a caracteres
          2
c
```

Repare que, nos dois últimos exemplos, em virtude do uso do %c, só foi listado um caractere de cada cadeia de caracteres passada como parâmetro. O valor 10 à frente do c não significa 10 caracteres. Um número seguindo o sinal de percentagem

Tabela 2: Sequências de escape

Sequência	Efeito
a	Soa o beep
b	Volta uma posição (<i>backspace</i>)
f	Salta para a próxima página lógica (<i>form feed</i>)
n	Salta para o início da linha seguinte (<i>line feed</i>)
r	Volta para o início da linha corrente (<i>carriage return</i>)
t	Avança para a próxima marca de tabulação

(%) significa o tamanho que a cadeia terá após a execução do comando. Vamos ver a seguir mais alguns exemplos. Os comandos abaixo:

```
$ printf "%d\n" 32
32
$ printf "%10d\n" 32
          32
```

preenchem a *string* com espaços em branco à esquerda (oito espaços mais dois caracteres, 10 dígitos), não com zeros. Já no comando abaixo:

```
$ printf "%04d\n" 32
0032
```

O 04 após % significa “formate a string em quatro dígitos, com zeros à esquerda se necessário”. No comando:

```
$ printf "%e\n" $(echo "scale=2 ; 100/6" | bc)
1.666000e+01
```

O padrão do %e é seis casas decimais. Já no comando:

```
$ printf "%.2e\n" `echo "scale=2 ; 100/6" | bc`
1.67e+01
```

O .2 especificou duas casas decimais. Observe agora:

```
$ printf "%f\n" 32.3
32.300000
```

O padrão do %f é seis casas decimais. E no comando:

```
$ printf "%.2f\n" 32.3
32.30
```

O .2 especificou duas casas decimais. Agora observe:

```
$ printf "%.3f\n" `echo "scale=2 ; 100/6" | bc`
33.330
```

O bc devolveu duas casas decimais e o printf colocou o zero à direita. O comando a seguir:

```
$ printf "%o\n" 10
12
```

Converteu o valor 10 para base octal. Para melhorar experimentalmente:

```
$ printf "%03o\n" 27
033
```

Assim a conversão fica com mais jeito de octal, né?. O que este aqui faz?

```
$ printf "%s\n" Peteleca
Peteleca
$ printf "%15s\n" Peteleca
          Peteleca
```

Imprime Peteleca com 15 caracteres. A cadeia de caracteres é preenchida com espaços em branco à esquerda. Já no comando:

```
$ printf "%-15sNeves\n" Peteleca
Peteleca      Neves
```

O menos (-) colocou espaços em branco à direita de Peteleca até completar os 15 caracteres pedidos. E o comando abaixo, o que faz?

```
$ printf "%.3s\n" Peteleca
Pet
```

O .3 manda truncar a cadeia de caracteres após as três primeiras letras. E o comando a seguir:

```
$ printf "%10.3sa\n" Peteleca
          Peta Pet
```

Imprime a cadeia com 10 caracteres, truncada após os três primeiros, concatenada com o caractere a (após o s). E esse comando a seguir, o que faz?

```
$ printf "EXEMPLO %x\n" 45232
EXEMPLO b0b0
```

Ele transformou o número 45232 para hexadecimal (b0b0), mas os zeros não combinam com o resto. Experimente:

```
$ printf "EXEMPLO %X\n" 45232
EXEMPLO BOBO
```

Assim disfarçou melhor! (repare no X maiúsculo). Pra terminar, que tal o comando abaixo:

```
$ printf "%X %XL%X\n" 49354 192 10
COCA COLA
```

Este aí não é marketing e é bastante completo, veja só como funciona:

O primeiro %X converteu 49354 em hexadecimal, resultando em COCA (leia-se “cê”, “zero”, “cê” e “a”). Em seguida veio um espaço em branco seguido por outro %XL. O %X converteu o 192 dando como resultado C0 que com o L fez COL. E finalmente o último parâmetro %X transformou o número 10 na letra A.

Conforme vocês podem notar, a instrução é bastante completa e complexa. Ainda bem que o echo resolve quase tudo...

Acertei em cheio quando resolvi explicar o printf através de exemplos, pois não sabia como enumerar tantas regrihas sem tornar a leitura enfadonha.

Principais variáveis do Shell

O Bash possui diversas variáveis que servem para dar informações sobre o ambiente ou alterá-lo. São muitas e não pretendo mostrar todas elas, mas uma pequena parte pode lhe ajudar na elaboração de scripts. Veja a seguir as principais delas:

CDPATH » Contém os caminhos que serão pesquisados para tentar localizar um diretório especificado. Apesar dessa variável ser pouco conhecida, seu uso deve ser incentivado por poupar muito

trabalho, principalmente em instalações com estrutura de diretórios em múltiplos níveis. Veja o exemplo a seguir:

```
$ echo $CDPATH
.:.:~:/usr/local
$ pwd
/home/jneves/LM
$ cd bin
$ pwd
/usr/local/bin
```

Como /usr/local estava na minha variável \$CDPATH e não existia o diretório bin em nenhum dos seus antecessores (., .. e ~), o comando cd foi executado tendo como destino /usr/local/bin.

HISTSIZE » Limita o número de instruções que cabem dentro do arquivo de histórico de comandos (normalmente .bash_history, mas na verdade é o que está indicado na variável \$HISTFILE). Seu valor padrão é 500.

HOSTNAME » O nome do host corrente (que também pode ser obtido com o comando uname -n).

LANG » Usada para determinar o idioma falado no país (mais especificamente categoria do locale). Veja um exemplo:

```
$ date
Thu Apr 14 11:54:13 BRT 2005
$ LANG=pt_BR date
Qui Abr 14 11:55:14 BRT 2005
```

LINENO » O número da linha do script ou função que está sendo executada. Seu uso principal é mostrar mensagens de erro juntamente com as variáveis \$0 (nome do programa) e \$FUNCNAME (nome da função em execução).

LOGNAME » Esta variável armazena o nome de login do usuário.

MAILCHECK » Especifica, em segundos, a frequência com que o Shell verifica a presença de correspondência nos arquivos indicados pela variáveis \$MAILPATH ou \$MAIL. O tempo padrão é de 60 segundos

(1 minuto). A cada intervalo o Shell fará a verificação antes de exibir o próximo prompt primário (\$PS1). Se essa variável estiver sem valor ou com um valor menor ou igual a zero, a busca por novas mensagens não será efetuada.

PATH » Caminhos que serão pesquisados para tentar localizar um arquivo especificado. Como cada script é um arquivo, caso use o diretório corrente (.) na sua variável \$PATH, você não necessitará usar o comando ./script para que o script script seja executado. Basta digitar script. Este é o modo que prefiro.

PIPESTATUS » É uma variável do tipo vetor (array) que contém uma lista de valores de códigos de retorno do último pipeline executado, isto é, um array que abriga cada um dos \$? de cada instrução do último pipeline. Para entender melhor, veja o exemplo a seguir:

```
$ who
jneves pts/0 Apr 11 16:26 (10.2.4.144)
jneves pts/1 Apr 12 12:04 (10.2.4.144)
$ who | grep ^botelho
$ echo ${PIPESTATUS[*]}
0 1
```

Neste exemplo mostramos que o usuário botelho não estava “logado”, em seguida executamos um pipeline que procurava por ele. Usa-se a notação [*] em um array para listar todos os seus elementos; dessa forma, vimos que a primeira instrução (who) foi bem-sucedida (código de retorno 0) e a seguinte (grep) não (código de retorno 1).

PROMPT_COMMAND » Se esta variável receber o nome de um comando, toda vez que você teclar um [ENTER] sozinho no prompt principal (\$PS1), esse comando será executado. É muito útil quando você precisa repetindo constantemente uma determinada instrução.

PS1 » É o prompt principal. No “Papô de Botequim” usamos os padrões \$ para usuário comum e # para root, mas é mui-

to freqüente que ele esteja personalizado. Uma curiosidade é que existe até concurso de quem programa o `$PS1` mais criativo.

PS2 » Também chamado “prompt de continuação”, é aquele sinal de maior (>) que aparece após um **[ENTER]** sem o comando ter sido encerrado.

PWD » Possui o caminho completo (`$PATH`) do diretório corrente. Tem o mesmo efeito do comando `pwd`.

RANDOM » Cada vez que esta variável é acessada, devolve um inteiro aleatório entre 0 e 32767. Para gerar um inteiro entre 0 e 100, por exemplo, digitamos:

```
$ echo $((RANDOM%101))
73
```

Ou seja, pegamos o resto da divisão do número randômico gerado por 101 porque o resto da divisão de qualquer número por 101 varia entre 0 e 100.

REPLY » Use esta variável para recuperar o último campo lido, caso ele não tenha nenhuma variável associada. Exemplo:

```
$ read -p "Digite S ou N: "
Digite S ou N: N
$ echo $REPLY
N
```

SECONDS » Esta variável informa, em segundos, há quanto tempo o Shell corrente está “de pé”. Use-a para demonstrar a estabilidade do Linux e esnobar usuários daquela coisa com janelinhas coloridas que chamam de sistema operacional, mas que necessita de “reboots” freqüentes.

TMOUT » Se esta variável contiver um valor maior do que zero, esse valor será considerado o *timeout* padrão do comando `read`. No prompt, esse valor é interpretado como o tempo de espera por uma ação antes de expirar a sessão. Supondo que a variável contenha o valor 30, o Shell encerrará a sessão do usuário (ou seja, fará *logout*) após 30 segundos sem nenhuma ação no prompt.

Expansão de parâmetros

Bem, muito do que vimos até agora são comandos externos ao Shell. Eles quebram o maior galho, facilitam a visualização, manutenção e depuração do código, mas não são tão eficientes quanto os intrínsecos (*built-ins*). Quando o nosso problema for performance, devemos dar preferência ao uso dos intrínsecos. A partir de agora vou te mostrar algumas técnicas para o seu programa pisar no acelerador.

Na **tabela 3** e nos exemplos a seguir, veremos uma série de construções chamadas expansão (ou substituição) de parâmetros (*Parameter Expansion*), que substituem instruções como o `cut`, o `expr`, o `tr`, o `sed` e outras de forma mais ágil.

Vamos ver alguns exemplos: se em uma pergunta o `S` é oferecido como valor *default* (padrão) e a saída vai para a variável `SN`, após ler o valor podemos fazer:

```
SN=$(SN:-S)
```

Para saber o tamanho de uma `cadeia`:

```
$ cadeia=0123
$ echo ${#cadeia}
4
```

Para extrair dados de `cadeia`, da posição um até o final fazemos:

```
$ cadeia=abcdef
$ echo ${cadeia:1}
bcdef
```

Repare que a origem é zero e não um. Vamos extrair 3 caracteres a partir da 2ª posição da mesma variável `cadeia`:

```
$ echo ${cadeia:2:3}
cde
```

Repare que novamente a origem da contagem é zero e não um. Para suprimir tudo à esquerda da primeira ocorrência de uma cadeia, faça:

```
$ cadeia="Papo de Botequim"
$ echo ${cadeia#*' '}
de Botequim
$ echo "Conversa "${cadeia#*' '}
Conversa de Botequim
```

No exemplo anterior foi suprimido à esquerda tudo o que “casa” com a menor ocorrência da expressão `*' '`, ou seja, todos os caracteres até o primeiro espaço em branco. Esses exemplos também poderiam ser escritos sem proteger o espaço da interpretação do Shell (mas prefiro protegê-lo para facilitar a legibilidade do código). Veja só:

```
$ echo ${cadeia#* }
de Botequim
$ echo "Conversa "${cadeia#* }
Conversa de Botequim
```

Repare que na construção de `expr` é permitido o uso de metacaracteres.

Utilizando o mesmo valor da variável `cadeia`, observe como faríamos para ter somente `Botequim`:

```
$ echo ${cadeia##*' '}
Botequim
$ echo "Vamos 'Chopear' no "${cadeia##*' '}
Vamos 'Chopear' no Botequim
```

Desta vez suprimimos à esquerda de `cadeia` a maior ocorrência da expressão `expr`. Assim como no caso anterior, o uso de metacaracteres é permitido.

Outro exemplo mais útil: para que não apareça o caminho (*path*) completo do seu programa (`$0`) em uma mensagem de erro, inicie o seu texto da seguinte forma:

```
echo Uso: ${0##*/} texto da mensagem de erro
```

Neste exemplo seria suprimido à esquerda tudo até a última barra (`/`) do caminho, restando somente o nome do programa. O caractere `%` é simétrico ao `#`, veja o exemplo:

Tabela 3: Tipos de expansão de parâmetros

Expansão de parâmetros	Resultado esperado
<code>\${var:-padrao}</code>	Se <code>var</code> é vazia, o resultado da expressão é padrão
<code>\${#cadeia}</code>	Tamanho de <code>\$cadeia</code>
<code>\${cadeia:posicao}</code>	Extrai uma subcadeia de <code>\$cadeia</code> a partir de posição. Origem zero
<code>\${cadeia:posicao:tamanho}</code>	Extrai uma subcadeia de <code>\$cadeia</code> a partir de posição com tamanho igual a <code>tamanho</code> . Origem zero
<code>\${cadeia#expr}</code>	Corta a menor ocorrência de <code>\$cadeia</code> à esquerda da expressão <code>expr</code>
<code>\${cadeia##expr}</code>	Corta a maior ocorrência de <code>\$cadeia</code> à esquerda da expressão <code>expr</code>
<code>\${cadeia%expr}</code>	Corta a menor ocorrência de <code>\$cadeia</code> à direita da expressão <code>expr</code>
<code>\${cadeia%%expr}</code>	Corta a maior ocorrência de <code>\$cadeia</code> à direita da expressão <code>expr</code>
<code>\${cadeia/subcad1/subcad2}</code>	Troca a primeira ocorrência de <code>subcad1</code> por <code>subcad2</code>
<code>\${cadeia//subcad1/subcad2}</code>	Troca todas as ocorrências de <code>subcad1</code> por <code>subcad2</code>
<code>\${cadeia/#subcad1/subcad2}</code>	Se <code>subcad1</code> combina com o início de cadeia, então é trocado por <code>subcad2</code>
<code>\${cadeia/%subcad1/subcad2}</code>	Se <code>subcad1</code> combina com o fim de cadeia, então é trocado por <code>subcad2</code>

```
$ echo $cadeia
Papo de Botequim
$ echo ${cadeia%' '*}
Papo de
$ echo ${cadeia%%%' '*}
Papo
```

Para trocar primeira ocorrência de uma subcadeia em uma cadeia por outra:

```
$ echo $cadeia
Papo de Botequim
$ echo ${cadeia/de/no}
Papo no Botequim
$ echo ${cadeia/de /}
Papo Botequim
```

Preste atenção quando for usar metacaracteres! Eles são gulosos e sempre combinarão com a maior possibilidade; No exemplo a seguir eu queria trocar *Papo de Botequim* por *Conversa de Botequim*:

```
$ echo $cadeia
Papo de Botequim
$ echo ${cadeia/*o/Conversa}
Conversatequim
```

A idéia era pegar tudo até o primeiro `o`, mas acabou sendo trocado tudo até o último `o`. Isto poderia ser resolvido de diversas maneiras. Eis algumas:

```
$ echo ${cadeia/*po/Conversa}
Conversa de Botequim
$ echo ${cadeia/????/Conversa}
Conversa de Botequim
```

Trocando todas as ocorrências de uma subcadeia por outra. O comando:

```
$ echo ${cadeia//o/a}
Papa de Batequim
```

Ordena a troca de todos as letras `o` por `a`. Outro exemplo mais útil é para contar a quantidade de arquivos existentes no diretório corrente. Observe o exemplo:

```
$ ls | wc -l
30
```

O `wc` põe um monte de espaços em branco antes do resultado. Para tirá-los:

```
# QtdArqs recebe a saída do comando
$ QtdArqs=$(ls | wc -l)
$ echo ${QtdArqs/ * /}
30
```

Nesse exemplo, eu sabia que a saída era composta de brancos e números, por isso montei essa expressão para trocar todos os espaços por nada. Note que antes e após o asterisco (*) há espaços em branco.

Há várias formas de trocar uma subcadeia no início ou no fim de uma variável. Para trocar no início fazemos:

```
$ echo $Passaro
quero quero
$ echo "Como diz o sulista - "${PassaroU
/#quero/não}
Como diz o sulista - não quero
```

Para trocar no final fazemos:

```
$ echo "Como diz o nordestino - "
"${Passaro/%quero/não}
Como diz o nordestino - quero não
```

– Agora já chega, o papo hoje foi chato porque teve muita decoreba, mas o que mais importa é você ter entendido o que te falei. Quando precisar, consulte estes guardanapos onde rabisquei as dicas e depois guarde-os para consultas futuras. Mas voltando à vaca fria: tá na hora de tomar outro e ver o jogo do Mengão. Pra próxima vez vou te dar moleza e só vou cobrar o seguinte: pegue a rotina `pergunta.func` (da qual falamos no início do nosso bate-papo de hoje, veja a [listagem 1](#)) e otimize-a para que a variável `$SN` receba o valor padrão por expansão de parâmetros, como vimos.

E não se esqueça: em caso de dúvidas ou falta de companhia para um (ou mais) chope é só mandar um e-mail para julio.neves@gmail.com. E diga para os amigos que quem estiver a fim de fazer um curso porreta de programação em Shell deve mandar um e-mail para julio.neves@tecnohall.com.br para informar-se. Valeu! ■

SOBRE O AUTOR

Julio Cezar Neves é Analista de Suporte de Sistemas desde 1969 e trabalha com Unix desde 1980, quando participou do desenvolvimento do SOX, um sistema operacional similar ao Unix produzido pela Cobra Computadores. Pode ser contatado no e-mail julio.neves@gmail.com

Lidando com a tecnologia proprietária

Planeta GNU

Adotar o Software Livre não significa ter de abrir mão de tecnologias populares. Vamos falar também dos perigos da Propriedade Intelectual.

POR CHRISTIANO ANDERSON

Quero começar a coluna deste mês homenageando o esforço do pessoal do Projeto Software Livre – Bahia (PSL-BA) para organizar o II Festival do Software Livre da Bahia [1], que foi realizado entre os dias 08 e 10 de Abril na linda cidade de Salvador. O evento foi muito bem organizado, uma oportunidade de encontro para desenvolvedores e amantes da liberdade de conhecimento. Diversos temas foram abordados, muitos deles técnicos, como *PyGTK*, e outros filosóficos, como o uso do Software Livre na educação, licenças e outros assuntos de interesse geral. Participei da última mesa redonda cujo assunto foi a Propriedade Intelectual, tema que foge dos limites técnicos e aborda um fator bastante social, já abordado anteriormente nesta coluna. Essa mesa foi composta por pessoas que defendem a liberdade de conhecimento, como o rapper BNegão [2], ex-vocalista do Planet Hemp. BNegão divulga suas músicas sob licença livre na Internet [3], de onde qualquer pessoa pode baixá-las e ouvi-las sem a necessidade de comprar o CD.

Propriedade Intelectual

O debate no Festival da Bahia foi enriquecido com as palavras de Cláudio Prado, do Ministério da Cultura, que apoiou totalmente a atitude de BNegão de divulgar seu trabalho para que todos possam apreciá-lo livremente. Isso mostra que os preceitos defendidos pelo Software Livre quebram a barreira da tecnologia e podem ser aplicados ao dia-a-dia de pessoas comuns.

A humanidade está passando por uma mudança muito importante. Antigamente as pessoas e empresas que tinham muito dinheiro eram as mais poderosas. Isso, hoje, não faz mais sentido – a alteração na consciência coletiva transformou o conhecimento na moeda mais valiosa atualmente. Percebemos que, quanto mais conhecimento agregado, maior o poder.

Vejam o exemplo da maior empresa de software do mundo, a Microsoft. Essa empresa conseguiu monopolizar o mundo da informática com seu sistema operacional, que hoje é o mais utilizado em computadores pessoais. Dessa forma, conseguiu arrecadar uma enorme fortuna, suficiente

para comprar qualquer outra empresa que venha a ameaçar seu monopólio. Na maioria das vezes, a empresa ameaçadora é adquirida e fechada, deixando assim o caminho livre para a continuidade do reinado da Microsoft. Com isso, nenhum outro sistema operacional jamais conseguiu atingir uma parcela significativa no mercado das estações de trabalho (e houve muitas tentativas, todas em vão).

O que eles não esperavam é que indivíduos estivessem se organizando, muitas vezes “na calada da noite”, para compartilhar seus conhecimentos e criar algo novo, diferente. Foi assim que nasceu o Software Livre, organizado por pessoas de todos os cantos do planeta que contribuem com o que de melhor sabem fazer e que cedem com gosto seu conhecimento para o bem coletivo. O Projeto GNU fornece os meios para que esse conhecimento não seja ferido e, muito menos, utilizado de forma parasítica em benefício das grandes corporações. A criação de licenças livres é o “X” da questão.



Imagem original: NevraX Design Team

O que a Microsoft pode fazer com a comunidade do Software Livre? Mesmo com todo o dinheiro que possui ela não pode comprar as pessoas e seus conhecimentos, não pode afetar toda uma comunidade. A grande arma da Microsoft é chamada de “Patentes de Software”. “Vamos patentear o máximo que a gente puder”, é o que pensam os executivos da empresa. De certa forma estariam, assim, dificultando o desenvolvimento do Software Livre. Dificultar não significa impedir. Uma comunidade unida sempre encontrará meios legais de continuar com seu magnífico trabalho.

Java Livre, Flash Livre

A indústria de software encontrou meios de seduzir usuários com softwares proprietários, criando recursos de áudio e vídeo que só são executados mediante a instalação de plugins e aplicativos não-livres, como o *Flash*, criado pela Macromedia (agora Adobe) e a máquina virtual *Java*, criada pela Sun. Embora gratuitos, esses dois aplicativos não são livres. Muitas vezes os usuários não sabem desse detalhe. Algumas distribuições de GNU/Linux os incorporam ao sistema por padrão, muitas vezes sem informar as restrições ligadas a cada um deles.

Java

O que muita gente não sabe é que existem alternativas livres para os dois aplicativos. Se você gosta de programar em Java, está convidado a conhecer um projeto mantido pela *Free Software Foundation*. Dividido em duas partes, ele é composto pelo *GNU Compiler for Java* [4] (GCJ), que nada mais é que um compilador Java que segue os preceitos de liberdade da FSF, e o *GNU Classpath* [5], projeto que reúne as classes Java mais comuns, mas de forma livre. O GNU Classpath ainda está em fase de desenvolvimento e precisa de ajuda para crescer. Se você tem bons conhecimentos de Java e quer compartilhar seu

conhecimento de forma livre, essa é a oportunidade que faltava para juntar as duas coisas. Ajude o projeto!

No Brasil existe o *Javali* [6], grupo de usuários criado para fortalecer o desenvolvimento do Java no Brasil e que também atua na divulgação de JVMs (Java Virtual Machines) livres e na criação de Software Livre com Java. Há brasileiros envolvidos nas partes mais críticas do desenvolvimento de tecnologia Java para todo o mundo.

Flash

Quando você está navegando na Internet, muitas vezes se depara com sites que foram feitos usando uma tecnologia proprietária da Macromedia, o Flash. O plugin, apesar de gratuito, não é livre. Ao instalá-lo, não sabemos o que está sendo executado em nosso computador, que tipo de informações ele coleta em nossa máquina etc. Infelizmente alguns sites só abrem se o usuário tiver o Flash instalado em sua máquina, obrigando-o assim a instalar esse aplicativo.

O projeto *GPLFlash* [7] nasceu para acabar com esta dificuldade. Apesar de ainda não ter todos os recursos suportados pelo plugin proprietário (como sons e interatividade), já é um grande começo. Com os avanços no desenvolvimento de projetos como este, em breve o usuário poderá usufruir de todos os recursos sem ser obrigado a instalar uma versão proprietária em sua máquina. ■

INFORMAÇÕES

[1] festival.softwarelivre.org

[2] www.bnegao.com.br

[3] www.midiaindependente.org/pt/red/2004/06/283492.shtml

[4] gcc.gnu.org/java/index.html

[5] www.gnu.org/software/classpath/

[6] www.javali.org.br

[7] gplflash.sourceforge.net/

Uma olhada atualizada no Software Livre e seus assecclas

Projetos na incubadora

O conhecido programa de configuração do SuSE, o YaST2, ajudará usuários do Debian no futuro. E se você está procurando um modo de relaxar ao término de suas incumbências de administrador de sistema, dê uma olhada em “Battle for Wesnoth.”

POR MARTIN LOSCHWITZ



É raro que a seção *Projetos na Incubadora* fale sobre algum jogo. A razão para isso é a falta de jogos novos e ao mesmo tempo interessantes para Linux. Para expiar essa falta de atenção dada aos jogos, este mês vamos nos concentrar num jogo de estratégia chamado *The Battle for Wesnoth* (figura 1). O jogo é leve o bastante para rodar em hardware mais antigo e oferece um projeto interessante que compensa muito bem a falta de gráficos 3D sofisticados.

Falarei também dos esforços do Debian para incorporar o YaST2, originalmente desenvolvido para o SUSE LINUX, e descreverei o sistema de arquivos SSHFS. Terminamos com uma novidade sobre o Debian Sarge.

* Estratégia de jogo

A rainha de Wesnoth está tramando contra seu marido com o objetivo de colocar um amiguinho no trono. Para isso, ela selou pactos secretos com os membros da corte real. O mago real, que é leal ao rei, descobre o complô e se volta contra

a rainha para salvar o herdeiro legítimo do trono. Se você se lembra de ter jogado *Warlords* nos bons tempos do Amiga, certamente vai se sentir em casa em *The Battle for Wesnoth*.

Há dois modos típicos para jogos de estratégia. O modo *realtime* (tempo real) significa que o jogo continua mesmo se não houver jogadores, enquanto o modo *round-based* (em turnos) espera que os jogadores façam suas jogadas. O *Wesnoth* usa esse segundo modo. Diferente de muitos outros jogos de estratégia, o jogo inicia com uma coleção completa de edifícios e, portanto, não se espera que os jogadores se envolvam em projetos de construção.

Colírio para os olhos

No mapa que mostra o reino de *Wesnoth* fica óbvio que os desenvolvedores colocaram bastante amor e carinho na criação do som e dos gráficos. Assim, o *Wesnoth* escapa do destino de muitos jogos em que um enredo ou jogabilidade excelentes sofrem com gráficos mal-desenhados. Bons gráficos não fazem um bom jogo sozinhos, mas ajudam.

Uma grande variedade de personagens e grupos, incluindo arqueiros e cavalaria com diversas características, contribuem para interessantes batalhas. Algumas unidades lutam melhor à noite, outras durante o dia. Os soldados têm diferentes níveis de energia vital, custos de manutenção e mesmo de inteligência. A inteligência é expressa pela velocidade com a qual as tropas aprendem com sua experiência na batalha. Xamãs cuidam dos feridos.

O jogo diferencia combate de curta e de longa distância. Se estiver na sua vez, você também deve escolher o tipo

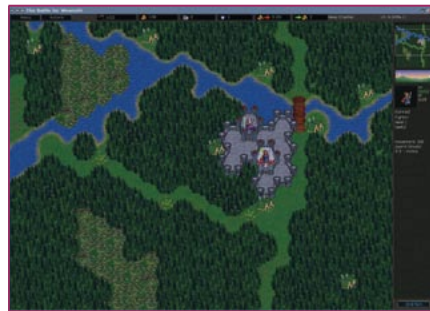


Figura 1: Na Batalha por Wesnoth, os jogadores têm que proteger a corte e o país de uma rainha do mal.

de armas que usará no ataque. A defesa é automática, desde que a unidade que esteja sofrendo o ataque tenha os recursos necessários. Se a unidade não possuir uma arma para combate corpo-a-corpo, pode estar condenada. Como em muitos outros jogos, o objetivo é apossar-se de tantas edificações quanto possível e derrotar o inimigo!

Um tutorial ajuda os jogadores a entrar no cenário de pesadelo do combate medieval e demonstra as etapas mais complicadas. The Battle for Wesnoth é frugal em suas exigências de hardware e software; necessita apenas da biblioteca SDL. E se você não gostar dos mapas que vêm com o jogo, pode ligar o editor de mapas e desenhar o seu próprio. Interessou? É só olhar a homepage [1] para maiores informações. ■

* YaST 2 sem SUSE

O YaST 2 é provavelmente o programa de configuração que, sozinho, dividiu a comunidade Linux mais do que qualquer outro ao longo dos últimos anos (figura 2). Essa invenção da SUSE foi um espinho no pé da comunidade do FOSS (*Free and Open Source Software* – Software Livre e de Código Aberto), já que não era desenvolvido sob uma licença livre e, por isso, contradiz a filosofia básica subjacente ao Linux. Ao mesmo tempo, o YaST 2 oferece ao usuário uma interface gráfica intuitiva, que é apreciada por especialistas e recém-chegados.

Os que ainda não usaram o SUSE LINUX até agora, não importa a razão, tiveram de se virar sem o YaST 2. Sem uma licença livre, não havia como portar o YaST 2 para outras distribuições. Felizmente, essa situação mudou. Em seguida à aquisição da SUSE, a Novell anunciou em março de 2004 que lançaria o YaST 2 sob uma licença livre.

A explicação dada pela Novell faz sentido: o YaST 2 da Novell oferecerá uma interface de configuração uniforme para

todos os sistemas Linux. Nos meses que se seguiram à remoção das restrições de licenciamento, não houve qualquer reação óbvia. Mas em novembro de 2004 Mario Fux começou a pensar alto sobre portar o YaST 2 na lista de discussão Debian-desktop. Afinal de contas, o Debian é carente em software de configuração. Enquanto o Mandrake e o Fedora têm diversas ferramentas com interface gráfica para incumbências administrativas críticas, os usuários do Debian precisam recorrer ao console. O Debian carece até de ferramentas em modo texto para algumas tarefas, o que força os usuários a abrir o editor de textos e modificar diretamente os arquivos de configuração, em alguns casos.

Mario Fux surpreendeu a audiência com um anúncio interessante. Disse que gostaria de dar uma recompensa e doar hardware para o desenvolvedor que portasse o YaST2 para o Debian. Fux deixa sua motivação bastante clara; pretende fazer do Debian um sistema operacional para usuários desktop, aconteça o que acontecer. Os desenvolvedores do Debian estão desde então discutindo os prós e os contras de se portar o YaST. Gerenciamento de software é visto como

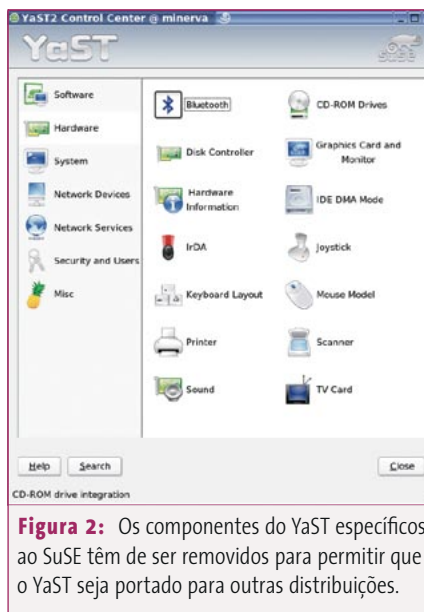


Figura 2: Os componentes do YaST específicos ao SuSE têm de ser removidos para permitir que o YaST seja portado para outras distribuições.

um assunto delicado, já que o SUSE e, portanto, o YaST 2, são baseados em pacotes RPM.

Um pequeno grupo de desenvolvedores formou-se para apoiar Mario Fux; atualmente esse grupo se ocupa de portar o YaST. Uma sessão de *chat* no início de fevereiro revelou as últimas notícias. O planejamento está num estágio bastante avançado e é provável que o projeto esteja seriamente encaminhado num futuro próximo. Mario Fux contactou o principal desenvolvedor do YaST 2 no SUSE, Klaus Kaempf, para ajudar a acelerar as coisas. O primeiro objetivo é criar um pacote Debian do YaST 2 para facilitar o trabalho de desenvolvimento. Uma homepage está sendo montada para coordenar as atividades dos desenvolvedores no futuro.

O grupo de desenvolvimento de Mario Fux descobriu uma peculiaridade: internamente, o YaST 2 usa uma linguagem de programação proprietária, a *YaST Control Language* (YCP), para todos os módulos. Isso dificulta mais ainda o problema. Além de remover elementos específicos do SUSE do YaST 2, os desenvolvedores precisarão aprender uma nova linguagem de programação que nenhum outro projeto usa.

Embora nenhum outro programa de configuração facilite tanto a vida dos recém-chegados ao Linux quanto o YaST, usuários com conhecimentos especializados se queixam de que ele não lhes oferece margem suficiente para alterações manuais. Todavia, usuários de computadores de mesa normalmente não querem ter esse tipo de controle. Se o grupo for bem-sucedido em portar o conjunto completo de recursos, o YaST 2 pode alavancar a popularidade do Debian GNU/Linux como um sistema operacional para desktop. A única fonte de informações sobre o projeto de portagem no momento é a lista de discussão *Debian-Desktop* [2]. ■

* Sistemas de arquivos no espaço do usuário

Se você já teve de mover um arquivo de um computador para outro através de uma rede, já está familiarizado com os problemas associados à transferência de arquivos: FTP é uma possibilidade, mas é inseguro. SCP é seguro, mas muito pouco amigável. Se for necessário apagar ou renomear um arquivo, você vai precisar de um login SSH.

Após abrir a sessão, pode-se rodar ferramentas familiares e convenientes como *cp*, *mv* e *rm*. Porém, SSH e SCP não permitem a execução remota desses comandos. Uma opção seria acrescentar comandos para apagar ou mover arquivos, como com o SCP. Porém, pelo lado ruim, os usuários precisariam decorar e digitar a sintaxe completa do comando para cada operação.

Miklos Szeredi parece ter se enchedo disso e desenvolveu uma nova solução. Nos velhos tempos do LUFSS (*Linux User File System*), ele usava o sistema de arquivos SSHFS, que simplesmente lhe permitia usar SSH como um driver de sistema de arquivos. Após montar o sistema de arquivos remoto usando SSHFS, os usuários tinham acesso a comandos típicos de shell. Infelizmente, o mantenedor do LUFSS desistiu em pouco tempo e não se encontrou um sucessor. Como o Linux 2.6 se transformava rapidamente e o LUFSS necessitava de um patch no kernel, o sistema era inútil.

Além disso, o projeto do LUFSS tinha alguns pontos fracos que o tornavam difícil de manter. E ainda por cima, já havia uma alternativa a caminho: o *Fuse* (*Filesystem in Userspace*) é muito parecido com o LUFSS do ponto de vista do usuário, embora use um método completamente diferente. Miklos Szeredi teve a idéia de reimplementar o LUFSS-SSHFS baseado no *Fuse* e removendo os erros do projeto durante o processo. O resultado é também chamado de SSHFS.

E como o *Fuse* funciona? Em primeiro lugar, é preciso um kernel que permita o uso de *Fuse*. O *Fuse* não é um sistema de arquivos de verdade, mas uma estrutura que pode carregar drivers distintos do espaço do usuário. O driver do *Fuse* no kernel faz, portanto, a tradução entre o kernel e o driver do espaço do usuário. Superficialmente, o produto terminado tem a mesmíssima aparência de um sistema de arquivos normal; operações fundamentais, como ler, copiar ou apagar arquivos podem ser realizadas com as ferramentas normais do espaço do usuário. Uma das maiores vantagens do *Fuse* é que usuários sem privilégios de root podem montar o sistema de arquivos. Para isso, precisam carregar o módulo do *Fuse* no kernel.

Testado e aprovado

O SFTP também pode lidar com gerenciamento de arquivos em sistemas de arquivos remotos. Porém, como as operações ocorrem dentro de uma sessão do SFTP, o transporte de dados entre os sistemas remoto e local é mais complicado que com o SSHFS e as ferramentas padrão *cp*, *mv* e *rm*. Graças ao SSHFS, os usuários podem facilmente rodar scripts e outras ferramentas automatizadas no sistema remoto.

Outra vantagem do SSHFS é o suporte a *multi-threading*. Isso permite que tarefas múltiplas sejam realizadas consecutivamente. O SSHFS faz o trabalho mais leve, especialmente com tarefas de cópia recorrente que envolvam o SFTP ou o SCP.

A estrutura do *Fuse* é responsável principalmente por facilitar a integração de uma sessão SSH. Mas as opções são mais ou menos infinitas. Por exemplo, os usuários podem montar o serviço Gmail do Google via GmailFS, que também é baseado no *Fuse*, exatamente como um compartilhamento de arquivos por NFS. Dessa forma os usuários podem usar os dois gigabytes (desde primeiro de abril)

de espaço oferecido pelo serviço para algo mais interessante. O desempenho do sistema de arquivos *Fuse* não chega àquilo que se esperaria de um sistema de arquivos de rede local ou com um fim especial; porém, desde que você tenha uma conexão rápida e estável, o *Fuse* pode ser uma opção útil. ■

* Instalador do Debian e DevFS

A próxima versão do Debian GNU/Linux, codinome *Sarge*, não vai precisar de disquetes de boot para ser instalado. Em vez disso, conterà o novo *Debian Installer* (figura 3) com uma arquitetura completamente nova e muito mais robusta. Joey Hess publicou em janeiro deste ano a versão do instalador que será incluída no *Sarge*. Mas agora está parecendo que o aparentemente infinito processo de lançamento, que postergou o surgimento do *Sarge* por meses, começou a ficar sedento por vingança.

O principal problema é o *DevFS*. Essa solução para um diretório */dev* organizado e desobstruído, que remonta aos tempos do Linux 2.4, caiu em desgraça junto aos desenvolvedores há um bom tempinho. Erros de projeto tornam o código difícil de manter. O Linux 2.6 tem uma nova abordagem: o *Udev*. Embora o *DevFS* ainda faça parte do kernel atual, há uma corrente que quer removê-lo para todo o sempre. Por exemplo, Linus Torvalds anunciou que ele seria removido do kernel em meados de 2005. Isso dará aos distribuidores uma oportunidade de substituir o antigo *DevFS* pelo *Udev* nesse meio tempo.

Isso afeta o instalador do Debian, que instalará o Linux 2.6 se necessário e habilitará o *DevFS* ao fazê-lo. Mudar isso exigiria um trabalho de muito baixo nível, que Joey Hass já declarou estar fora de questão. Isso significa mais uma trapalhada dos desenvolvedores do Debian: o *Sarge* será baseado num software obsoleto e não-suportado quando for finalmente

lançado. Embora isso pareça ser inofensivo à primeira vista – afinal de contas, as versões estáveis do Debian não usam o kernel mais recente – será preciso aplicar patches de desenvolvimento se ocorrerem problemas de segurança.

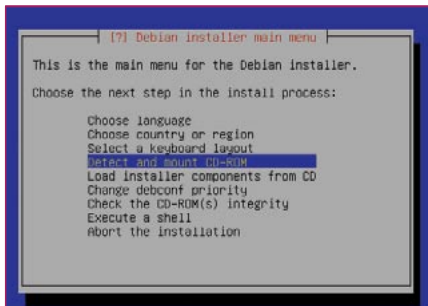


Figura 3: Problemas com o instalador do Debian: ele ainda usa o DevFS, um projeto órfão que deve ser removido do kernel num futuro próximo.

E esse é o maior risco do DevFS, já que não há mantenedor oficial para ele. Tanto as partes do DevFS no kernel quanto aquelas no espaço do usuário ficaram órfãs já há um bom tempo. Ou seja, a equipe de segurança do Debian precisaria de fato manter o DevFS para remover erros críticos ou remediar problemas de segurança. Isso significa um monte de trabalho, já que o Sarge terá muito mais pacotes do que a atual versão *Woody*; por isso, o fardo carregado pela Equipe de Segurança deverá aumentar consideravelmente. ■

* O caminho para o Sarge...

Problemas desse tipo só podem ser evitados no futuro diminuindo-se drasticamente os intervalos entre as versões do Debian. Mesmo assim, não há qualquer sinal de um lançamento do Sarge, que está num estado mais ou menos indefinido há meses. Numa mensagem à lista de discussão *Debian-Devel-Announce* [2] há algumas semanas, Steve Langasek – membro da equipe de lançamento do Debian – anunciou que diversos problemas graves haviam sido resolvidos na versão de teste graças aos novos pacotes GCC, KDE 3.3 e

Perl. Ele também mencionou muito trabalho duro para aumentar a segurança do Sarge e que havia hardware disponível para as arquiteturas permitidas.

...é longo e pedregoso

Isso é como uma luz no fim do túnel, uma vez que a falta de suporte à segurança tem sido vista como um enorme obstáculo nos últimos tempos. Para dar apoio a essa maré positiva, outra *Bug Squashing Party* (algo como *Festa do Inseticida*) foi realizada na primeira semana de fevereiro e os participantes foram bem-sucedidos em remover mais problemas. Se você está interessado em ler a mensagem de Langasek, na qual ele também comenta sobre as etapas futuras, dê uma olhada nos arquivos da lista de discussão em [3].

Os membros da equipe de segurança já estão habituados ao problema do software obsoleto. Está cada vez mais difícil encontrar patches de segurança para o Debian Woody. A versão atual tem mais de dois anos – e novos patches são com frequência extremamente difíceis de instalar. Se forem precisos mais alguns anos até que surja o sucessor do Sarge, a questão é como a Equipe de Segurança poderá continuar a carregar o DevFS nas costas. ■

* Isso é tu-tu-tudo, pe-pe-pessoal...

... ao menos neste mês. Se você quiser recomendar um programa que gostaria de ver nesta seção, que tal me mandar um email com sua sugestão [4]? Aguardo ansiosamente seus comentários! ■

INFORMAÇÕES

[1] The Battle for Wesnoth: www.wesnoth.org

[2] Lista de discussão Debian-Desktop: lists.debian.org/debian-desktop

[3] Steve Langasek sobre atualizações no Sarge: lists.debian.org/debian-devel-announce/2005/01/msg00011.html

[4] Dicas e sugestões (em inglês): projects@linux-magazine.com

CALENDÁRIO DE EVENTOS

EVENTO	DATA	LOCAL	WEBSITE
FISL 6.0	1 a 4 de Junho	Porto Alegre, RS	fisl.softwarelivre.org
LinuxCorp	05 e 06 de Julho	São Paulo, SP	www.rpmbrazil.com.br
What the Hack	28 a 31 de Julho	Liempde, Holanda	www.whatthehack.org
XXIII ENECOMP	01 a 05 de Agosto	Bonito, MS	www.enec.org.br
Wikimania 2005	04 a 08 de Agosto	Frankfurt, Alemanha	wikimania.wikimedia.org
LinuxWorld Conference & Expo 2005	09 a 11 de Agosto	San Francisco, EUA	www.linuxworldexpo.com
III CONISLI	10 de Novembro	São Paulo, SP	www.conisli.org.br
III SOLISC	01 de Dezembro	Florianópolis, SC	www.softwarelivresc.org.br
OpenBeach 6.0	03 de Dezembro	Florianópolis, SC	www.softwarelivresc.org.br/openbeach
II Latinoware	27 de Março, 2006	Foz do Iguaçu, PR	www.latinoware.org

FISL 6.0

Data: 1 a 4 de Junho de 2005
Local: Porto Alegre, RS
Website: fisl.softwarelivre.org

LinuxCorp

Data: 5 e 6 de Julho de 2005
Local: São Paulo, SP
Website: www.rpmbrazil.com.br

III CONISLI

Data: 10 de Novembro de 2005
Local: São Paulo, SP
Website: www.conisli.org.br

ESCREVA PARA A LINUX MAGAZINE

Estamos sempre à procura de novos artigos e autores. Se você acha que um assunto é importante, ou que precisa ser melhor explicado, fale conosco.

Precisamos de tutoriais, análises, estudos de caso e notícias. Se você é membro de um grupo de usuários, porque não nos conta sobre os eventos que estão sendo planejados? Preferimos que os artigos sejam enviados via e-mail, e screenshots são sempre bem-vindos. Para facilitar as coisas, mencione no assunto de sua mensagem o tema do artigo.

Artigos têm em média 4.500 caracteres por página (contando os espaços), mas listagens de código e imagens reduzem o espaço disponível para o texto. Se possível, escreva páginas inteiras. Como estamos presentes em vários países, evite o uso de gírias e expressões regionais.

As imagens devem estar na maior resolução possível. No caso de fotos digitais, recomendamos que elas sejam tiradas com uma câmera de 3 Megapixels e resolução igual ou superior a 1024x768 pixels. Prefira formatos como TIFF e JPEG (em qualidade *High*) para fotografias, PNG para ilustrações e formatos vetoriais como SVG, AI e EPS para logotipos.

Uma revista passa por muitos estágios durante sua produção, e alguns meses podem se passar desde que seu artigo seja recebido até que a revista chegue às bancas. Portanto, nunca envie na última hora material ou notícias sobre encontros e eventos.

Envie suas colaborações para o endereço material@linuxmagazine.com.br. Evite enviar mais de 4 MB em arquivos anexos. Caso o material necessário para seu artigo ultrapasse esse limite, coloque-o em algum site na Internet e nos informe a URL.

Os Monges DANIEL TREZUB



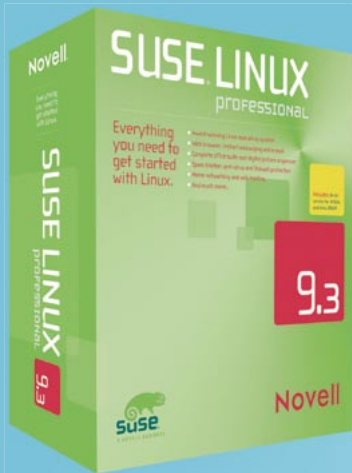
ÍNDICE DE ANUNCIANTES

SERVIÇOS

ÍNDICE DE ANUNCIANTES		
ANUNCIANTE	SITE	PÁGINA
A Casa do Linux	www.casadolinux.com.br	96
Alternativa Linux	www.alternativainux.com.br	73
Atípico	www.atipico.com.br/curso	95
Conisli	www.conisli.org.br	91
Dell	www.dell.com.br	100 (quarta-capa)
Ermida	www.ermida.com.br	96
FISL	fisl.softwarelivre.org	99 (3ª capa)
Green Treinamento e Consultoria	www.green.com.br	09
Infomedia	www.infomediatv.com.br	19
Linux Corp	www.rpmbrazil.com.br	51
Linux New Media	www.linuxnewmedia.com.br	02 (contra-capa), 71
Linux World	www.linuxworldexpo.com	63
NC Software	www.ncsoftware.com.br	17
OTC	ead.otcinfo.com.br	35
Unicial / LinuxPress	www.unicial.com.br	77
utah	www.utah.com.br	96

Julho de 2005: décima edição

Na próxima edição...



Camaleão de roupa nova

A última versão do SUSE LINUX traz as versões atualizadas dos aplicativos que compõem a distribuição, vários módulos do YaST (o “Painel de Controle” do SUSE) melhorados e outras mudanças significativas, além de novidades como o sistema de busca *Beagle* e o software para catalogação de imagens *F-Spot*. Veja porque uma atualização do sistema é uma boa pedida .

Imprensa On-Line

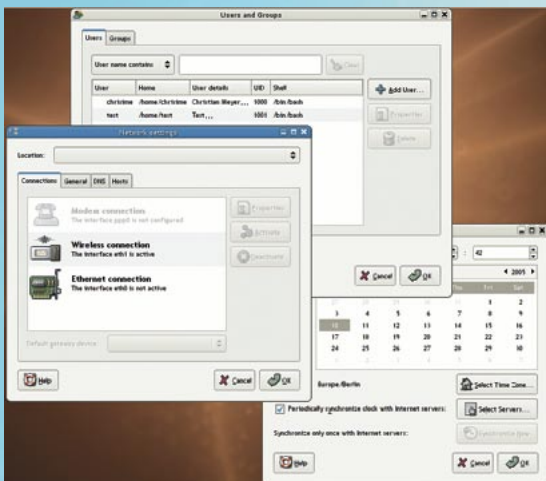
O tema da próxima edição trata das ferramentas que permitem o trabalho cooperativo e colaborativo através da rede, mesmo para aqueles que não sejam desenvolvedores web: os “wikis”, “blogs” e gerenciadores de conteúdo em geral. Você conhecerá o *MediaWiki*, a ferramenta por trás da famosa enciclopédia colaborativa Wikipedia, que já se tornou uma referência para o trabalho colaborativo via web. Com ela, usuários podem adicionar notas a documentos, revisá-los e complementá-los, desfrutando de controle de revisão e de acesso “de primeira”. Falaremos também do *TWiki* e mostraremos como usá-lo para criar a Intranet da sua empresa ou o seu site pessoal. Além disso, você vai aprender como criar uma comunidade e um blog usando o *Drupal*, e conhecer as principais novidades do *Zope X3.0*, a mais recente versão do Zope, um popular e poderoso gerenciador de conteúdo para sites na Internet escrito na versátil linguagem Python.

GMailFS

O servidor do GMail é realmente generoso: o sistema disponibiliza 2 GB de espaço para emails a seus usuários. Mas por que deixar todo esse espaço livre dando sopa até que ele encha, quando você pode usá-lo como um sistema de arquivos acessível via Internet? A edição de julho da Linux Magazine mostra detalhadamente como transformar a sua conta no GMail em um disco rígido virtual.

Gnome 2.10

Aparentemente, a nova versão do Gnome traz uma variedade de novos aplicativos e *applets*. Mas por debaixo do capô, a versão 2.10 traz um sistema muito mais robusto. Nosso artigo vai mostrar as modificações mais importantes introduzidas pelo Gnome 2.10 – muitas das quais não são apenas recursos inacessíveis ao usuário comum, mas novidades que vão realmente “encher os olhos” de todo mundo.



CD Número 09

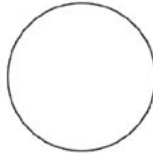
junho 2005

LINUX

MAGAZINE

PRODUZIDO
NO POLO INDUSTRIAL
DE MANAUS
CONHEÇA A AMAZÔNIA

COMPACT
disc



Aviso: Este CD-ROM da Linux Magazine foi testado extensivamente e até onde pudemos verificar se encontra livre de qualquer vírus ou outro tipo de software de conteúdo malicioso, bem como de defeitos. A Linux Magazine não se responsabiliza por qualquer dano ou perda de dados que advinha da utilização deste CD-ROM ou de software nele incluído. Este CD é parte integrante da revista Linux Magazine nº09 e não pode ser vendido separadamente.

gentoo

2005.0

**A distribuição mais rápida
e atualizada do planeta!**

Produzido no polo Industrial de Manaus por Sopropress Rimo da Amazônia Ind. e Com. Fonográfica LTDA - Indústria Brasileira - CNPJ: 84.494.129/0001-93 - sob encomenda de Linux Near Media LTDA - 06.351.942/0001-45

Produzido no polo Industrial de Manaus por Sopropress Rimo da Amazônia Ind. e Com. Fonográfica LTDA - Indústria Brasileira - CNPJ: 84.494.129/0001-93 - sob encomenda de Linux Near Media LTDA - 06.351.942/0001-45