



ENFEITANDO O DRAGÃO
KDE com um gostinho de Mac OS X



PAPO DE BOTEQUIM:
Expressões, sinais e parâmetros

TECNOLOGIA SEM LIMITES

LINUX MAGAZINE

NÚMERO 10

PUBLICANDO NA WEB

MEDIAWIKI

TWIKI

DRUPAL

ZOPE

SUSE LINUX

GNOME

PROGRAMAÇÃO SEGURA

KOOLDOCK

KXDOCK

APOLLON

JULHO 2005

NÚMERO 10

LINUX
MAGAZINE

LINUX

MAGAZINE



**DRUPAL,
MEDIAWIKI,
ZOPE, PLONE!**

LINUX/NEW/MEDIA
The Pulse of Linux

PUBLICANDO NA WEB

O poder do MediaWiki
TWiki: colaboração via web na empresa
Instalando e configurando o Drupal
As novidades do Zope X3.0

O camaleão voltou

Analizamos o SUSE LINUX 9.3

Segurança

Tape os buracos de seus programas e scripts

Veja também:

Transforme o Gmail em um disco virtual
Apollon: troca de arquivos no KDE
O que mudou no Gnome 2.10
Estratégias de backup

R\$14,90
€ 6,50
Ed. 10
07/2005



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, Augusto Campos, Carsten Schnober,
 Christian Meyer, Christiano Anderson, Dominik Vogt,
 Elisabeth Bauer, Fabrizio Ciacchi, Hagen Höpfner, Julio Cezar
 Neves, Marc Andre Selig, Marcel Hilzinger, Peter Thoeny,
 Philipp von Weitershausen, Rafael Peregrino da Silva, Ricardo
 Dalceno, Uwe Hermann.

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 advenham 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.

Distribuído por Fernando Chinaglia Distribuidora.

Direitos Autorais e Marcas Registradas © 2004:
 Linux New Media do Brasil Editora Ltda.

Impressão e Acabamento: Editora Gráficos Burti LTDA

ISSN 1806-9428

Impresso no Brasil



em processo de filiação

Déjà vu tecnológico...

Prezado leitor, prezada leitora da Linux Magazine,

dá-se o nome de *Déjà vu* à sensação de que algo visto pela primeira vez parece ser inexplicavelmente familiar.

No início eram os *mainframes*. E os gerentes de TI, que naquela época nem sabiam que tinham esse nome, achavam bom. E o software era livre. Todo o processamento era concentrado em uma única máquina de grande porte, ligada a terminais burros. Caso mais capacidade de processamento fosse necessária, era preciso trocar o *mainframe* inteiro. A isso se convencionou chamar “escalabilidade vertical”. Apesar de tudo, *mainframes* continuam populares nos dias de hoje, graças a técnicas de virtualização do hardware através das quais múltiplos sistemas operacionais podem ser executados em paralelo na mesma máquina de grande porte.

E então vieram os minicomputadores e, com eles, as redes de computadores. Estas, por sua vez, resultaram na introdução de paradigmas interessantes, como o famoso “O computador é a rede”. O advento do PC levou ao barateamento, à conseqüente “comoditização” do hardware para microcomputadores e à possibilidade de criar *clusters*.

Clusters são conjuntos de computadores de baixo custo, conectados a uma rede de alto desempenho e processando tarefas de modo cooperativo. São tão poderosos quanto *mainframes*, mas apresentam três grandes vantagens: baixo custo com a mesma capacidade de processamento, escalabilidade horizontal/incremental (caso você precise de mais capacidade de processamento, basta introduzir mais PCs ao *cluster*) e alta disponibilidade. Mas nem tudo são rosas no mundo dos *clusters*: eles são, geralmente, um desafio de gerenciamento.

Para resolver o problema de espaço que *clusters* tendem a apresentar, foi criada uma estrutura de *racks* em que o hardware é acondicionado de forma especialmente compacta: as “pizza boxes” de 1,75 polegadas de altura (1 unidade de *rack* ou 1U). Até 44 máquinas nesse formato podem ser acondicionadas em um *rack* padrão. Elas formam os blocos básicos de uma estrutura que se convencionou chamar Data Center.

Mas a coisa não pára por aí: servidores *blade* são a promessa para aliar o gerenciamento simples dos *mainframes* à flexibilidade e escalabilidade dos *clusters*. O hardware consiste em PCs modulares montados em uma placa-mãe única com todos os seus periféricos. Esses módulos (*blades*) são montados em um chassis de 4U de altura dotado de todas as conexões necessárias para a operação do sistema (alimentação, rede e gerenciamento). Cada chassis pode conter entre 15 a 20 *blades*, o que significa dizer que um único *rack* pode abrigar um *cluster* com até 200 PCs!

A impressão que fica é de que já vimos tudo isso antes: ora são PCs altamente integrados em servidores *blade* que se comportam como *mainframes*, ora são *mainframes* com centenas de sistemas operacionais rodando em paralelo, se comportando como PCs. Entretanto, qualquer que seja a sua escolha, o Linux e os sistemas de código aberto em geral estão predestinados a rodar em todas essas plataformas. A bem da verdade, as tecnologias abertas são as únicas que já rodam em todas as plataformas descritas e elas vão desempenhar um papel fundamental no futuro próximo da sua estrutura de TI. *Déjà vu*? É bom ficar por dentro...



Rafael Peregrino da Silva
 Editor



Cartas

06

Notícias

Sem surpresas

Conversamos com o CEO da Mandriva sobre os resultados da fusão entre a Mandrakesoft e a Conectiva.

O pingüim precisa aprender a abrir o bico

Não basta vender Linux. Quem não se comunica não sobrevive no mercado de tecnologia.

Mundo livre em revista

Dicas de [In]segurança

Notícias do Kernel

Gestão brasileira com Linux

Veja o que o vice-presidente de tecnologia da Microsiga pensa do Linux e do Software Livre.

07
07

08

10

14

16

18

Capa

A praça é do povo!

Esforço comunitário

Conheça o *MediaWiki*, sistema por trás da *Wikipedia*.

Construtor de equipes

TWiki, um excelente software colaborativo para o mundo empresarial.

Conteúdo a conta-gotas

Mostramos a instalação e configuração do *Drupal*.

Fórmula X

Veja o que há de novo no *Zope X3.0*.

21

21

22

28

35

42

Análises

Camaleão à prova – 9.3

Saiba o que mudou na nova versão do *SUSE LINUX*.

Seguindo as pegadas

Gnome 2.10: mais aplicativos, mais robusto e inovador.

46

46

52

Tutoriais

Dados pra viagem

Transforme sua conta no *Gmail* em um disco virtual.

57

57

21

A praça é do povo!

No início dos tempos, a web era uma via de mão única. Alguma entidade paranormal se esgueirava por detrás dos servidores, estalava os dedos e, *presto!* O conteúdo aparecia automaticamente, esculpido com cuidado para o deleite dos internautas – mas ainda estático, perene, imutável. Mas os internautas queriam mais: hoje eles querem ser, também, deuses da informação e divulgar, para quem queira conhecer, suas próprias idéias e opiniões. O tema de nossa revista deste mês é a popularização da Internet e da fácil publicação de conteúdo – por qualquer pessoa – por meio de ferramentas que facilitam a vida do usuário não-técnico: o *Wiki* e o *Blog*.

Essas ferramentas levam o conceito de cooperação e colaboração na Web a níveis jamais vistos. E o melhor de tudo: ninguém precisa de doutorado em engenharia aeroespacial para usá-las. Mostraremos como um usuário comum de Linux pode configurar rápida e facilmente um site colaborativo usando as mesmas ferramentas livres e de código aberto que os especialistas amam.

46

Camaleão à prova – 9.3

A mais recente versão do *SUSE LINUX Profissional* foi lançada em meados de abril. Como era de se esperar, a caixinha traz uma porção de programas antigos, alguns novos e um monte de módulos *YaST* aperfeiçoados, além de muita documentação. Descubra o que mais mudou no *SUSE 9.3* (além, é claro, das letras maiúsculas no nome).

57

Dados pra viagem

O *Gmail*, sistema de *webmail* do Google, foi festejado como o novo paradigma do correio eletrônico via Internet. Uma conta do *Gmail* vem com um espaço para guardar mensagens que pode chegar aos 2 GB. Entretanto, oferecer gratuitamente uma grande quantidade de espaço de armazenamento a um usuário de Linux abre as portas da criatividade em certas cabecinhas privilegiadas, levando à imediata inovação: o *GmailFS* permite usar os gigabytes de sua conta no *Gmail* como se fosse um diretório compartilhado em outra máquina da rede.

68 Poluição do ambiente

A maioria dos administradores já ouviu falar dos famosos estouro de buffer e *cross site scripting* ou de ataques envolvendo a formatação de cadeias de caracteres (*string format*). Existem vários livros com explicações sobre a causa destas falhas e como evitá-las. Ao contrário do que se poderia imaginar, esse conhecimento não é importante apenas para os desenvolvedores de software. Os administradores, também, devem saber como escrever (sob pressão) scripts seguros e como evitar falhas comuns ao modificar programas.

Mesmo quando a solução passa longe da teoria que aprendemos na faculdade ou a eficácia deixa a desejar, ninguém deve arriscar o surgimento de falhas de segurança. Muitas vezes a regra é: desativar o programa e fechar as lacunas. Nesse caso o administrador deverá avaliar o que é realmente necessário e se uma atualização de correção (*patch*) resolverá mesmo o problema. Este artigo, primeira parte de uma pequena série em dois atos, tem a intenção de transmitir os conhecimentos específicos necessários.

84 Revezamento olímpico

O *Kazaa* é a mais popular rede de compartilhamento de arquivos na Internet. Entretanto, até pouco tempo os usuários Linux estavam de fora da festa, pois não havia uma versão do cliente oficial compatível com o pingüim. A coisa mudou de figura com o projeto *GiFT* (*Gnu Internet File Transfer*), um daemon que implementa o acesso à rede *FastTrack*, usada pelo *Kazaa*, e várias outras rede *peer-to-peer* populares, como a *OpenNap* (do finado *Napster*) e *Gnutella* (usada pelo *Limewire*). Conheça o *Apollon*, um cliente *GiFT* que permite que você acesse, baixe e compartilhe arquivos em várias redes ao mesmo tempo, com um programa só.

93 Centro das atenções

Os últimos meses foram pródigos em eventos em que o Linux foi, se não o principal, um dos temas que mais atraiu a atenção de seus visitantes. De eventos especializados, passando por eventos da comunidade voltados para grupos de usuários até eventos governamentais, todo mundo quer saber o que a baiana, digo, o pingüim tem.



63
63

SysAdmin

Coluna do Augusto

Planeje como gerenciar o conteúdo de seu site.

64

E aí, tem becape?

Os prós e contras das principais estratégias de backup.

68
68

Programação

Poluição do ambiente

Truques que todo administrador deve conhecer para tornar seus scripts e programas mais seguros.

79
80

Linux User

CD do Mês

Em nosso CD, você encontrará os principais programas mencionados nesta edição.

82

Trabalhando nas docas

Deixe seu KDE com um "gostinho" de Mac OS X.

84

Revezamento Olímpico

Acesse as mais populares redes *peer-to-peer* no Linux.

86

Papo de Botequim – Parte X

Vamos aprender a avaliar expressões, capturar sinais e a receber parâmetros através da linha de comando.

92
92

Comunidade

Planeta GNU

Vamos relembra alguns passos importantes na criação de nossa *cybercultura*, como a ARPANET e o início da Internet.

93

Centro das atenções

Saiba o que aconteceu nos eventos de Software Livre nos últimos meses.

96
96
97
98

Serviços

Eventos / Mercado Linux

Quadrinhos / Anunciantes

Na próxima edição

Cartas para o editor

Permissão de escrita

✎ Dupla personalidade

» Excelente o artigo *Dupla Personalidade* publicado na quarta edição da Linux Magazine. Ele foi realmente valioso na minha volta ao mundo dos video-games. Depois de ler a matéria comprei logo um joystick *Twin Shock Vibration* da Clone (sugerido no artigo) e comecei a brincar principalmente com o *ePSXe 1.6.0* e o *ZSNES*.

Gostaria de saber se vocês têm referências mais específicas relativas a emuladores e joypads no Linux, principalmente informações sobre como otimizar configurações e extrair o máximo possível do hardware e software.

Estou tentando encontrar alguma informação sobre a possibilidade de configurar no Linux a vibração para meu joystick USB. Isso já é possível? Também estou querendo comprar um segundo joystick para jogos multiplayer e também gostaria de saber se vocês têm alguma recomendação.

Eduardo Sztokbant

Olá, Eduardo! Não entendemos exatamente o que você quis dizer com "otimizar configurações". Pelo que levantamos na Internet, a vibração de seu joystick não é suportada no Linux, e o *ePSXe* também não suporta os manches analógicos, mesmo com plugins extras como o *AmoQ's PadJoy*. Essa é uma falha do *ePSXe*, já que os manches analógicos funcionam perfeitamente em outros emuladores, como o excelente *MAME* (www.mame.net). Quanto a um segundo joystick, se você gostou do *Twin Shock*, por que não mais um do mesmo modelo?

Uma fonte de informações sobre o assunto (com links para outras páginas relevantes) é a página de *Vojtech Pavlik*, autor do driver de joystick usado no Linux. O endereço é: atrey.karlin.mff.cuni.cz/~vojtech/joystick ■

✎ WiMax

» Fiquei espantado pelo fato de a análise da matéria *O alfabeto 802.11*, publicada na sétima edição, não falar sobre a mais promissora das tecnologias wireless, a *WiMax*. Não falar dela é simplesmente ignorar o futuro.

Sidney Moura Lima.

Brasília – /DF

Decidimos mencionar no artigo apenas as tecnologias e produtos mais populares no mercado e os já padronizados de acordo com o IEEE (Institute of Electrical and Electronics Engineers, dos EUA). Os produtos WiMax disponíveis no mercado trabalham com uma versão preliminar do padrão, e as primeiras unidades de acordo com a especificação final só devem chegar ao mercado na segunda metade deste ano. Mais informações sobre essa tecnologia, que promete taxas de transferência de até 70 Mb/s sem linha de visão direta com a antena, podem ser obtidas no site do WiMax Forum, em www.wimaxforum.org. ■

✎ Música e Pingüins

» Sou novato no mundo Linux, mas estou muito entusiasmado com o que vejo. Minha esposa trabalha com música, e eu gostaria de saber se existe algum programa que imprima partituras, como o *Encore* ou o *Finale* no Windows.

Ottmar Landenberger

São Paulo

O melhor programa para compositores é o *GNU LilyPond* (lilypond.org), que usa uma linguagem própria para descrever as partituras. Infelizmente ele não tem uma interface gráfica, mas felizmente há softwares que usam o *LilyPond* como base e resolvem esse problema, como o *Rosegarden* (um ambiente para edição e

composição de música, encontrado em www.rosegardenmusic.com e o *NoteEdit*, disponível em notedit.berlios.de.

✎ Bancos de Dados

» Como desenvolvedor, sinto muito a falta de comparativos de desempenho sérios entre bancos de dados Open Source rodando no Linux. Será que não haveria um espaço numa das próximas edições para isso? Gostaria de saber a verdade sobre as várias opções, como o *MySQL*, *Firebird*, *PostgreSQL*, etc.

Hernan Nardelli Fonseca

Hernan, já pode reservar nossa próxima edição com seu jornalista, pois nossa equipe na Alemanha preparou uma série de artigos sobre o tema. Falaremos sobre o *PostgreSQL 8*, *Clusters com MySQL*, *Backup online e ajuste fino de servidores Oracle* e uma ferramenta para conversão de dados entre os "dialetos" SQL usados pelos vários SGBDs (Sistemas Gerenciadores de Banco de Dados) encontrados no mercado. ■

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



CEO da Mandriva fala da situação atual da fusão

Sem surpresas

Quatro meses após o anúncio oficial da fusão entre a Mandrakesoft e a Conectiva, a Mandriva vai muito bem, obrigado. A Linux Magazine conversou com François Bancelhon, CEO da empresa, sobre a situação atual do processo de fusão.

POR RAFAEL PEREGRINO DA SILVA

Linux Magazine» Depois de quatro meses do anúncio oficial, qual o balanço que vocês fazem do processo de integração das duas empresas?

François Bancelhon» O processo de aquisição já está efetivamente fechado, sem surpresas. Tudo está se correndo como deveria, apesar de uma fusão de duas empresas não ser uma coisa simples – pelo contrário: são, em geral, uma coisa difícil.

LM» Como está sendo a integração?

FB» Em muitos pontos, a integração está sendo mais fácil do que eu imaginava. A razão para isso está no modelo de desenvolvimento do Software Livre: os nossos desenvolvedores já se conheciam, já conheciam mais ou menos quais contribuições haviam sido dadas por uns e por outros, etc. Além disso, são duas equipes de desenvolvimento de altíssimo nível. Desse modo, do ponto de vista técnico, tenho certeza de que as duas equipes estão capacitadas e devem criar um produto unificado que contenha o melhor dos dois mundos. Isso deve ocorrer entre setembro e novembro deste ano. No que tange às equipes de serviço, elas vão estar presentes nas áreas em que as duas empresas atuavam. Podemos dizer que teremos uma equipe global de serviços. A proximidade geográfica é muito importante para o cliente, de modo que nós dispensamos atenção especial a isso. No final teremos uma integração total das equipes de desenvolvimento, controle de qualidade, suporte, serviços e vendas, respeitando as características de cada um dos mercados em que estamos atuando.

LM» Como a fusão foi recebida nos dois países?

FB» Na França o negócio foi muito bem recebido. A maioria dos nossos clientes e o próprio mercado que já faz uso de tecnologia aberta na França já conhecia a Conectiva. No Brasil, ao que eu saiba, a reação em geral não foi diferente – mesmo porque a Conectiva desfrutou de um reforço de caixa com a aquisição. Afinal, com a fusão ela se tornou parte de um conglomerado de abrangência global em sua área de atuação. Como os “estilos” das duas empresas são muito complementares, os pontos fracos de cada uma delas estão sendo suprimidos pela excelência técnica da outra. Por uma questão de foco, a Conectiva sempre foi forte em serviços e a Mandrakesoft em produtos. Agora somos fortes e competitivos nessas duas frentes, capazes de fornecer serviços e produtos de alta qualidade. Aliás, isso tem servido para alavancar novos projetos e negócios em um âmbito mundial. Além disso, afinamos o nosso discurso junto ao cliente e temos uma visão ainda mais clara do trinômio ROI (*Return on Investment* – planos que mostram em quanto tempo o cliente pode recuperar o dinheiro investido em um novo projeto), qualidade e treinamento.

Creio que um dos motivos da boa acolhida da fusão, tanto por parte do mercado quanto por parte da comunidade, se deve ao fato de que se tratou de uma fusão amigável, ou seja, um casamento de interesses, voltado para o crescimento mútuo. Não é o caso de uma empresa estar “assumindo” o controle da outra. O próprio nome “Man-

driva” é um atestado disso. Muito mais, se trata de criar um organismo novo, mais forte e robusto para encarar o mercado e os seus desafios para os próximos anos.

LM» Trabalhar em um mercado global tem sido mais difícil ou estressante?

FB» Mais interessante! É lógico que novos processos estão sendo criados para dar conta de um fluxo de informações mais dinâmico e que as coisas agora tomam uma outra proporção, já que a empresa cresceu repentinamente. A Mandriva tem que estar pronta para a evitar *overheads* e duplicidade de esforços, uniformizar estratégias etc. Os novos processos deverão dar cabo de todos esses problemas e “lubrificar as engrenagens” da nova companhia.

LM» Qual foi o impacto criado pela fusão nos negócios das duas empresas?

FB» Tanto no Brasil quanto na França, nossos clientes estratégicos foram informados do que estava por acontecer antes da fusão e receberam a notícia muito bem. Considero que o impacto da fusão para os negócios e clientes das “duas” empresas tenha sido extremamente positivo. Não gastamos mais dinheiro com isso, já que o negócio ocorreu via troca de ações. Com a fusão nosso desempenho melhorou e estamos capitalizando a empresa com novos recursos. Tanto a Mandrakesoft quanto a Conectiva viveram situações financeiras difíceis há pouco tempo, mas eram estáveis e aprenderam com os erros do passado.

LM» Quais são os desafios à frente?

FB» Há vários, mas alguns são fundamentais. A finalização do processo de fusão é um deles. Isso toma tempo e, como eu já disse anteriormente, nunca é fácil. Continuar crescendo externamente permanece, também, um desafio – para o qual, entretanto, devemos estar bem preparados. Além disso, é importante manter o foco na missão central da empresa. Isso é primordial para o nosso sucesso. ■

Quem não se comunica...

O pingüim precisa aprender a abrir o bico

Não basta apenas trabalhar com Linux e oferecer bons serviços. Se as empresas não souberem comunicar as vantagens desta tecnologia a seus clientes em potencial, não conseguirão crescer e sobreviver ao mercado.

POR ALEXANDRE BARBOSA



por Alfredo & Sonia K.
www.sxc.hu

Não creio que possa me considerar um ferrenho defensor do Linux como alguns colegas, mas até que, no geral, tenho uma postura pró-Linux devido às possibilidades que a plataforma oferece quanto à computação de qualidade com baixo custo – isso pode ajudar o Brasil a queimar etapas no processo de desenvolvimento tecnológico, além de ajudar na inclusão digital.

É portanto com um grau elevado de decepção que vejo o amadorismo das empresas brasileiras “dedicadas” ao Linux na hora de divulgarem seus produtos ou serviços. A coisa começa com o marketing sofrível das poucas companhias estabelecidas que não sabem definir seu foco de atuação. E antes que os trolls de plantão comecem com a

ladainha, é até desculpável que pequenas empresas construídas em torno do Linux atirem para todos os lados em busca de fontes de faturamento, dando cursos, certificando, gerenciando implementações ou desenvolvendo software. Mas quando se pergunta a essas empresas o que elas fazem melhor, a resposta é o costumeiro “tudo”.

Que é o mesmo que “nada”, quando se quer conquistar um cliente, principalmente os executivos que assinam os cheques. Lembre-se de que os executivos confiam partes de seu negócio apenas aos melhores da área desejada. Não é por acaso: eles precisam confiar em bons fornecedores e é difícil alcançar isso em empresas que se propõem a fazer “tudo” (isso em qualquer campo, mesmo não sendo tecnologia).

Outra falha é que as empresas precisam aprender a propor soluções e não tecnologias. Quem trabalha com Linux pode amar a plataforma, mas as pessoas que vão usá-lo não precisam, necessariamente, fazer o mesmo. É necessário, portanto, apresentar os projetos de forma mais profissional, concentrando-se em itens como desempenho nas operações, estabilidade, segurança e redução de custos. Não são essas as principais virtudes do ambiente? Sim, são. Então vamos nos ater a elas.

Onde está o pulo do gato, então? As empresas querem as vantagens, mas não desejam necessariamente saber se e qual Linux estará em suas redes. De novo, a batalha pelo avanço do Linux nas empresas brasileiras deve ser uma decisão de negócios e não uma escolha tecnológica. Que a batalha de “corações

e mentes” fique do lado dos desenvolvedores. Para fazer uma analogia, é como o consumidor que quer um carro com motor X *flex-fuel*, direção hidráulica, ar-condicionado e afins, sem entender patavina de mecânica. Como a indústria automobilística vende tanto se a média do consumidor pouco sabe além de onde se troca o óleo? É porque se concentram nos pontos sensíveis do consumidor em sua comunicação.

Voltando ao Linux, empresários (leia-se compradores) são sensíveis a palavras e expressões como “reduza seus custos”, “economize”, “leve vantagem”. São elas que devem ser trabalhadas nos materiais de divulgação das empresas de Linux, seja em brochuras, folhetos ou na Web. Igualmente importante é estabelecer casos de sucesso. São textos contando histórias de implementações bem-sucedidas em empresas, trazendo detalhes como os obstáculos vencidos, performance anterior e posterior à implementação e outras informações que poderiam ser de interesse de outros clientes potenciais.

Até aqui pareceu uma conversa sobre marketing, certo? É isso mesmo, porém mais ligado à definição do termo: *conjunto de estudos e medidas relacionados ao lançamento, promoção e distribuição de um produto ou serviço no mercado consumidor, visando à boa aceitação e ao sucesso comercial (Michaelis)*.

E é justamente por conta disso que as grandes corporações é que têm conseguido, paradoxalmente, ganhar mais dinheiro com o Linux. Confira aí na lista: IBM, CA, Sybase, Oracle e IBM, apenas para citar algumas que não são “pure Linux”. É óbvio que elas têm a seu favor uma história pregressa, recursos financeiros, tecnológicos e clientes existentes passíveis de migração. Mas também é verdade que sua abordagem usa e abusa de todas as ferramentas conhecidas de comunicação.

Ao chegar em clientes potenciais, contam com farto material como estudos de caso de empresas similares, mostram curvas de ROI (*Return On Investment* – planos que mostram em quanto tempo o cliente pode recuperar o dinheiro investido em um novo projeto), além de projeções de ganhos potenciais de produtividade, dando muito mais segurança a quem vai tomar uma decisão no rumo do Linux.

Já comentei o assunto com várias empresas brasileiras “pure Linux”, a maioria de médio porte, e fiquei espantado com as reações. Há quem acredite em produzir banners para eventos com a lista de coisas que pode fazer. Friso: o que PODE fazer e não o que realmente FAZ. Também vi reações agressivas, de quem acha que apontar essas deficiências é ir contra a plataforma e de que os desafios e pressões que essas pequenas empresas enfrentam já são grandes o suficiente sem ter de envolver ainda mais esse tipo de coisa. Sinto avisar que o buraco é mais embaixo.

Primeiro, porque não é o porte de uma empresa que a faz ser madura na comunicação de seus valores. É a visão estratégica de quem está no leme que alavanca o nome de uma companhia. Entre ficar depurando código por esporte e trocar mensagens em fóruns que só validam os preconceitos mais comuns, essas pessoas deveriam assistir ao “Pequenas Empresas, Grandes Negócios” ou buscar o apoio de entidades como o Sebrae, se necessário.

De novo: concordo em que o Linux seja uma tecnologia com atributos revolucionários e que, sim, ao trabalhar com Linux, qualquer um passa a fazer parte dessa revolução. Só não esperem condescendência do mercado ou das demais pessoas por conta disso. É melhor estar aparelhado para enfrentar a frieza corporativa com objetividade e contribuir para o progresso do Linux no meio empresarial brasileiro. ■

Mundo livre em revista

❑ Projeto Harmony: Proposta para uma versão livre do Java 5 na Fundação Apache



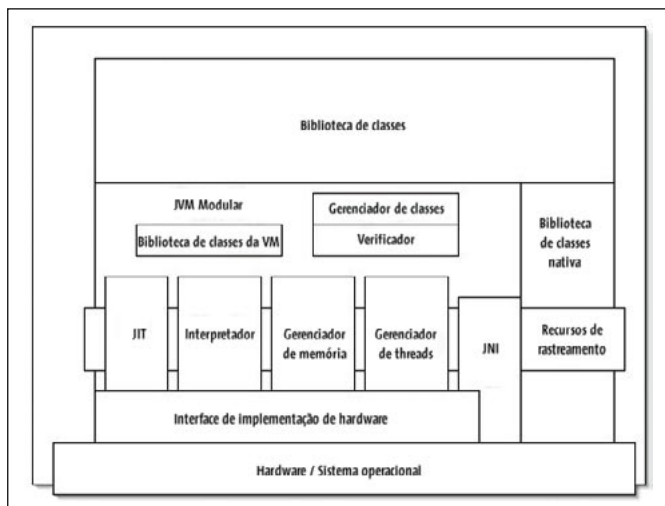
O incubador da Fundação Apache recebeu uma proposta de projeto para implementação de uma versão livre do Java 2 *Standard Edition* versão 5, compatível com a implementação oficial da Sun Microsystems (distribuída sob uma licença não-livre) e que deverá ser disponibilizada sob licença Apache v2.

A idéia, apresentada por Geir Magnusson, é criar uma arquitetura modular para a VM (máquina virtual) e a biblioteca de classes desenvolvida pela comunidade, de modo a permitir que implementações independentes possam

compartilhar seus componentes, assim como estimular a inovação individual desses componentes. Um diagrama da arquitetura modular proposta foi disponibilizado.

tinyurl.com/a5fyh

people.apache.org/~geirm/harmony.jpg



❑ Disponível nova versão estável do FreeBSD

Os desenvolvedores do projeto FreeBSD lançaram a versão 5.4 do seu Unix de código aberto. Como a série 5 do projeto havia sido declarada estável em novembro do ano passado (a partir do lançamento da versão 5.3), as novidades da versão atual não são muito visíveis. Além de algumas melhorias menores e muitas correções de segurança, bem como de erros de implementação críticos que afetavam a estabilidade do sistema, quase não há novos recursos no sistema.

O foco dessa versão foi notadamente a melhoria do suporte a máquinas multiprocessadas. Vários *drivers* foram modificados com esse propósito e o *scheduler* de processos, que estava desativado desde a versão 5.3 devido a um erro de implementação grave, está disponível novamente.

Apesar do FreeBSD ser um sistema de código aberto, são também disponibilizados *drivers* binários para ele. Um exemplo disso é o *driver* para o controlador RAID-5 de 8 portas SATA da empresa *High Point*, que faz parte do kernel FreeBSD padrão.

Uma importante novidade é a implementação do protocolo CARP, cujo desenvolvimento foi iniciado pelo projeto OpenBSD há cerca de dois anos. Com esse protocolo é possível criar soluções de alta disponibilidade para firewalls e servidores, que podem dividir o mesmo endereço IP. Além disso, algumas das ferramentas do OpenBSD foram incorporadas ao sistema. Mais detalhes podem ser encontrados no anúncio do lançamento da nova versão.

Nos dois CD-ROMs do FreeBSD estão disponíveis, entre outros, pacotes binários do X.org 6.8.2, KDE 3.4.0 e Gnome 2.10. A ordem do conteúdo dos CDs foi modificada a partir dessa versão. Ao contrário das versões anteriores, se encontram no primeiro CD, entre outras coisas, o sistema base, pacotes para o X.org e um sistema de recuperação, enquanto no segundo CD estão a documentação do sistema e grande parte dos pacotes binários. As imagens ISO dos CDs podem ser baixadas do servidor FTP do projeto ou de um de seus servidores espelho.

www.freebsd.org/

www.freebsd.org/releases/5.4R/announce.html

www.freebsd.org/releases/5.3R/announce.html

www.openbsd.org/

www.freebsd.org/releases/5.4R/relnotes.html

[ftp://ftp.freebsd.org/pub/FreeBSD/releases/i386/ISO-IMAGES/5.4](http://ftp.freebsd.org/pub/FreeBSD/releases/i386/ISO-IMAGES/5.4)

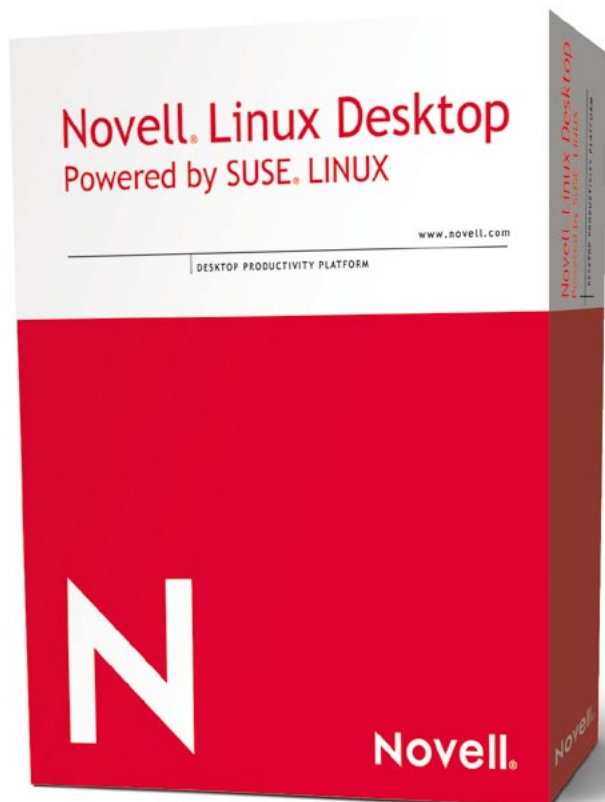
www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/mirrors-ftp.html

❑ Parceria entre a Novell e a Intel lança o Novell Linux Desktop no Brasil

No dia 16 de Maio, em uma coletiva de imprensa realizada em parceria com a Intel, a Novell anunciou o *Novell Linux Desktop 9* (NLD 9), a solução da empresa para estações de trabalho corporativas. Baseado no *SUSE LINUX Enterprise Server*, o produto da Novell inclui um conjunto de aplicativos para escritório (o *OpenOffice.org*, modificado pela empresa), aplicativo de email, calendário e colaboração (o *Evolution*) e uma versão atual do navegador *Firefox*, também customizado de forma a tirar o máximo de proveito do Linux no desktop. Ferramentas para design gráfico e multimídia também fazem parte desta distribuição Linux, apresentada à imprensa de modo extremamente profissional.

O Novell Linux Desktop 9 estará disponível por um preço sugerido de R\$162,50 – o que inclui mídia e atualizações por um ano – por meio dos parceiros de canal Novell.

Uma boa novidade oriunda do trabalho conjunto da Novell com a Intel é a certificação de hardware para o NLD. Várias das placas-mãe Intel para sistemas de mesa já receberam a certificação YES da Novell. As duas empresas estão trabalhando juntas para garantir que o sistema da Novell trabalhe bem mesmo com as tecnologias da Intel mais recentes, tais como *Intel Virtualization Technology*, *Intel Extended Memory 64*, *Intel Centrino Mobile™*, *Intel Dual-core Processors*, etc.



Segundo Ricardo Fernandes, presidente da Novell no Brasil, “a partir de agora o mercado corporativo dispõe de uma solução integrada madura de hardware e software para o desktop”. Para a empresa, o caminho natural para a migração de estações de trabalho está bem definido: ele se inicia em máquinas de uso mais restrito, tais como quiosques e estações de trabalho usadas como posições de atendimento de *telemarketing* em *call centers*, passando pelas estações de trabalho nos departamentos de engenharia das corporações e chegando finalmente às máquinas dos usuários avançados (*power users*). “A migração requer planejamento, preparação e execução”, justifica Fernandes.

Mas foi Oscar Clarke, presidente da Intel no país, quem fez a afirmação que causou maior furor durante a coletiva: “Cada vez mais vocês vão ouvir falar de Software Livre e de Linux, principalmente no ambiente governamental e corporativo, e menos de Windows. Essa é uma tendência inexorável!” Indagado se tal afirmação não poderia trazer problemas para a Intel junto à Microsoft, Clarke respondeu: “Em uma economia globalizada nós desenvolvemos nossas plataformas para atender à demanda do mercado – e essa demanda no Brasil é, cada vez mais, por Linux”. Diante da surpresa dos jornalistas na plateia, Clarke completou: “Além disso, os produtos da Microsoft também funcionam nos processadores da concorrência”.

Para divulgar a solução, a Novell e a Intel realizaram na segunda quinzena de maio uma série de seminários com duração de meio dia para clientes e parceiros em quatro grandes capitais brasileiras. ■

www.novell.com.br/

www.intel.com.br/

www.novell.com/pt-br/products/desktop/

❑ Sun Microsystems adquire Tarantella

A Sun Microsystems adquiriu a Tarantella por 25 milhões de dólares – o que corresponde a 90 centavos de dólar por ação da companhia. A Tarantella era uma antiga divisão da empresa *Santa Cruz Operations* (a SCO original) antes de esta ser vendida para a Caldera (empresa criadora da distribuição *Caldera OpenLinux*). É fornecedora de uma solução para uso remoto e seguro de ambientes desktop, aplicativos e dados, chamada *Secure Global Desktop*.

A aquisição necessita ainda da aprovação das comissões reguladoras do governo norte-americano (a *Securities and Exchanges Commission*) e, naturalmente, dos acionistas. O encerramento da transação é esperado para o final do ano fiscal de 2006. ■

www.sun.com/

www.tarantella.com/

❑ Nasce o PC Conectado!

Após quatro adiamentos desde o começo do ano, o governo federal finalmente anunciou, no dia 12 de Maio, o início do programa de isenção de impostos e de financiamento para microcomputadores a preços populares equipados com Software Livre: nasce o PC Conectado.

Com um preço sugerido entre 1.300 e 1.400 reais, o micro será financiado para os varejistas com recursos do BNDES (Banco Nacional de Desenvolvimento Econômico e Social) com base na Taxa de Juros de Longo Prazo (TJLP), desde que o sistema operacional e os aplicativos instalados sejam de código aberto. A expectativa, no entanto, é que o preço do PC Conectado fique bem abaixo desse limite, já que o governo deixará de cobrar o PIS/COFINS (Programa de Integração Social/Contribuição para o Financiamento da Seguridade Social) na venda para o consumidor final. Isso deverá representar uma queda potencial de 9,5% no custo do equipamento. Essa isenção de tributos vale também para qualquer outro desktop (computador de mesa), desde que custe até R\$ 2.500. A intenção de ampliar o benefício, segundo o assessor especial do Presidente da República, Cezar Alvarez, é combater o mercado de computadores pessoais montados com peças contrabandeadas.

O principal alvo do programa são famílias com renda entre três e dez salários mínimos com condições de destinar parte de sua renda à aquisição de um computador, além de pequenos e microempresários. As condições de financiamento para o consumidor final dependerão do varejista. “Os juros deverão variar entre 1% e 4,5%, dependendo da negociação do vendedor com o cliente. O vendedor determinará a prestação final ao consumidor”, afirma Alvarez, acrescentando que a prestação mensal deverá ficar em torno de R\$ 65 (para um financiamento em 24 meses).

De acordo com o anúncio, os provedores de Internet assumiram o compromisso de oferecer aos usuários do PC Conectado um pacote de 15 horas de acesso à Internet (por meio de linha discada) que custará R\$ 7,50 (com impostos).

Para facilitar o acesso à tecnologia, foi definida uma solução de suporte sem custo adicional, que contempla tanto os problemas relacionados à disponibilidade de serviços quanto os relacionados à falta do conhecimento necessário para a operação de microcomputadores. Duas categorias de suporte serão oferecidas àqueles que optarem pela aquisição do equipamento especificado no programa:

- **Suporte de disponibilidade** – suporte telefônico à instalação, configuração e customização do computador e de todos os aplicativos com ele oferecidos.
- **Suporte de funcionalidade** – suporte por email (ou a critério do fabricante) para atividades básicas relacionadas à operação do computador.

Além desse suporte ao usuário também será oferecida assistência técnica para problemas de hardware. A solução de suporte será oferecida pelo prazo mínimo de 1(um) ano, a partir da data da compra. A expectativa do governo é atingir a marca de 1 milhão de computadores vendidos neste ano e outro 1 milhão em 2006. “Temos absoluta confiança no sucesso do programa”, enfatiza Oscar Vaz Clarke, diretor-geral da Intel no Brasil, que colaborou para o programa com testes para garantir uma plataforma mínima de hardware que suportasse, de forma satisfatória, os softwares inclusos no pacote definido pelo governo.

A especificação do PC Conectado anunciada pelo governo é de um equipamento com processador de 1.5 GHz, 128 MB de memória RAM, disco rígido de 40 GB, fax/modem de 56 Kbps, placa de rede Ethernet, som e vídeo (*onboard*), monitor de 15 polegadas, teclado (no padrão ABNT2) e mouse. Entre os aplicativos estão um sistema operacional GNU/Linux, editor de textos, planilha eletrônica e gerador de apresentações, navegador de Internet, anti-vírus, *firewall* pessoal, cliente de email, mensageiro instantâneo, reproduzidor de arquivos multimídia etc. – no total, um pacote de 27 aplicativos, tudo baseado em Software Livre. ■

www.softwarelivre.gov.br/noticias/PC%20Conectado/view



❑ Lançada versão 3.7 do OpenBSD

A nova versão do OpenBSD já está disponível. Ela traz principalmente correções para erros de implementação e melhor suporte a hardware. Algumas atualizações de versão de software também foram realizadas: as versões do sistema operacional para as arquiteturas i386 e PowerPC usam o *gcc* 3.3.5. O servidor de janelas padrão do sistema é agora o X.org, disponível na versão 6.8.2 – por razões de compatibilidade o XFree86 3.3.6 também faz parte da distribuição. Correções de segurança também foram efetuadas na nova versão, tais como a que repara o problema de negação de serviço passível de ser gerado usando manipulação de *time stamps* em pacotes TCP.

Detalhes sobre as mudanças no OpenBSD 3.7 podem ser encontrados no *changelog* detalhado do projeto. Um resumo pode ser visto nas *Release Notes*. Modificações específicas para cada plataforma estão em suas respectivas páginas no site do projeto. Alterações na *ports collection*, a lista de software disponível da ferramenta de gestão de pacotes do projeto BSD, estão documentadas em uma página à parte.

As novas plataformas de hardware que podem ser utilizadas com essa nova versão são os PDAs da “família” Zaurus, da Sharp e a estação de trabalho MIPS de 64 bits SGI O2 – segundo os de-



envolvedores do projeto, versões do OpenBSD para outros sistemas da SGI estão sendo desenvolvidos.

O OpenBSD está disponível atualmente para 16 plataformas de hardware diferentes: de sistemas Intel de 32 bits, passando por máquinas AMD64, SPARC e PowerPC, até sistemas VAX; está sendo desenvolvido o suporte a outras arquiteturas, enquanto que, para outras, ele está sendo descontinuado.

O projeto OpenBSD é famoso pela segurança padrão de seus sistemas. Portabilidade, conformidade com padrões abertos, segurança e recursos de criptografia integrados ao sistema são os pontos fortes do trabalho de desenvolvimento. Em maio de 2003 a equipe de desenvolvimento do OpenBSD introduziu importantes conceitos de segurança contra estouros de memória (*buffer overflows*), como proteção contra *stack smashing* com o *StackGuard*. A versão 3.7 pode ser baixada via FTP, HTTP, AFS ou RSYNC da página do projeto na Internet ou dos diversos

servidores-espelho disponíveis.

www.openbsd.org/

www.kb.cert.org/vuls/id/637934

www.openbsd.org/37.html

www.openbsd.org/plat.html

www.openbsd.org/zaurus.html

www.openbsd.org/cgi.html

❑ Lançada a versão 2.0 do Free Pascal

Após 5 anos de desenvolvimento, foi lançada a versão 2.0 do compilador Pascal de código aberto *Free Pascal*, sucessora direta da versão 1.0. O *Free Pascal* é um compilador para sistemas de 32 e 64 bits compatível com o Turbo Pascal e, a partir da nova versão, também com o Delphi (até a versão 7, inclusive). A versão 1.0 estava disponível para os seguintes sistemas operacionais: DOS, Linux, Windows®, OS/2, FreeBSD, NetBSD e OpenBSD. A partir da versão 2.0 a lista de plataformas suportadas cresceu consideravelmente: Mac OS classic, Mac OS X, MorphOS e Novell Netware entraram para o rol dos sistemas operacionais para os quais o compilador está disponível oficialmente. No que tange aos processadores suportados, a partir de agora o compilador pode gerar código executável também para AMD64, SPARC, PowerPC (32 bits) e ARM.

Uma das principais novidades em relação à versão 1.0 é um melhor suporte a bancos de dados. Além disso, o compilador dispõe agora de suporte a *threads*, mais *units* de compatibilidade com o Delphi, ambiente de desenvolvimento integrado e suporte ao sistema de janelas em modo texto *Free Vision* (semelhante

ao antigo *Turbo Vision*, da Borland). O compilador está sob a Licença Pública Geral do Projeto GNU (*GNU General Public License* – a GPL). Já as bibliotecas de tempo de execução (*runtime libraries*) do sistema estão sob uma versão modificada da Licença Pública Menos Geral do Projeto GNU (*GNU Lesser General Public License* – a LGPL), que permite a sua utilização também para a confecção de programas comerciais. O compilador pode ser baixado do servidor FTP do projeto ou de um de seus servidores-espelho.

www.freepascal.org/

www.borland.com.br/delphi/

www.gnu.org/licenses/licenses.pt.html#GPL

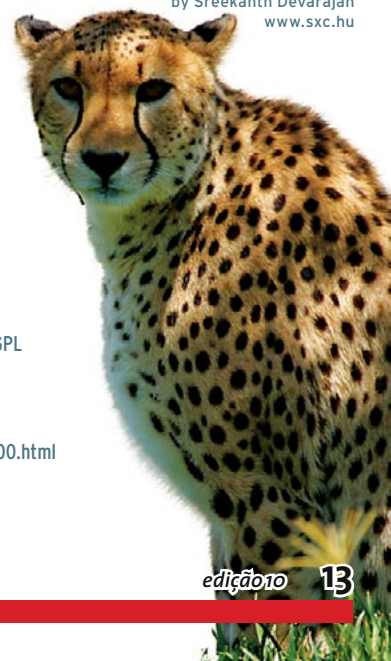
www.gnu.org/licenses/licenses.pt.html#TOCLGPL

[ftp://ftp.freepascal.org/fpc/dist/](http://ftp.freepascal.org/fpc/dist/)

www.freepascal.org/download.html

community.borland.com/article/0,1410,20803,00.html

by Sreekanth Devarajan
www.sxc.hu



Dicas de [In]segurança

❑ krb5

O Kerberos é um sistema de autenticação em rede que usa um terceiro elemento confiável, chamado KDC, para autenticar os clientes e servidores, fazendo as correspondências entre eles.

O pacote *krb5-workstation* inclui um cliente *Telnet* já preparado para autenticação via Kerberos. Dois estouros de buffer foram descobertos na maneira como o cliente de Telnet trata as mensagens vindas do servidor. Um invasor poderia executar código arbitrário na máquina da vítima se puder persuadi-la a conectar-se a um servidor Telnet malicioso. O projeto “Common Vulnerabilities and Exposures” (cve.mitre.org) deu a essa falha os códigos CAN-2005-0468 e CAN-2005-0469. ■

Referência no Debian: DSA-703-1 krb5

Referência no Gentoo: GLSA 200504-04 / telnet

Referência no Mandriva: MDKSA-2005:061

Referência no Red Hat: RHSA-2005:330-06

❑ MySQL

O MySQL é um servidor de banco de dados SQL multitarefa e multiusuário. Esta atualização repara inúmeras situações de risco no servidor MySQL.

Stefano Di Paola descobriu duas falhas na maneira como o MySQL manipula funções definidas pelo usuário. Um agressor poderia criar e executar funções definidas por ele próprio, levando à execução de código arbitrário no servidor MySQL. O projeto “Common Vulnerabilities and Exposures” (cve.mitre.org) deu a essa falha os códigos CAN-2005-0709 e CAN-2005-0710.

Di Paola também encontrou uma falha na maneira como o MySQL cria tabelas temporárias. Um usuário local poderia criar um link simbólico especialmente preparado de forma a ludibriar o MySQL, fazendo-o sobrescrever um arquivo para o qual tenha permissão de escrita.

O projeto “Common Vulnerabilities and Exposures” (cve.mitre.org) deu a essa falha o código CAN-2005-0711. ■

Referência no Gentoo: GLSA 200503-19 / mysql

Referência no Mandriva: MDKSA-2005:060

Referência no Red Hat: RHSA-2005:334-07

Referência no SuSE: SUSE-SA:2005:019

❑ Telnet

O pacote *telnet* oferece um cliente em modo texto para o protocolo Telnet. O pacote *telnet-server* inclui um *daemon* de um servidor de Telnet – o *telnetd* – que permite logins remotos na máquina em que roda.

Dois estouros de buffer foram encontrados no cliente, na rotina de tratamento de mensagens vindas do servidor. Um invasor poderia executar código na máquina da vítima se ela puder ser persuadida a conectar-se a um servidor Telnet malicioso. O projeto “Common Vulnerabilities and Exposures” (cve.mitre.org) deu a essa falha os códigos CAN-2005-0468 e CAN-2005-0469. ■

Referência no Debian: DSA-703-1 krb5

Referência no Gentoo: GLSA 200504-04 / telnet

Referência no Red Hat: RHSA-2005:327-10

❑ Mozilla

O Mozilla é um conjunto de programas de código aberto para a Internet que inclui um navegador Web, um poderoso cliente de email, um leitor de grupos de discussão (*newsgroups*), um programa para bate-papo em salas na Internet pelo protocolo IRC e um editor de HTML.

Há um estouro de buffer na maneira como o Mozilla processa imagens GIF. Um invasor poderia facilmente criar uma imagem GIF especialmente manipulada que, quando aberta dentro do Mozilla pela vítima, permitiria a execução de código arbitrário. O projeto “Common Vulnerabilities and Exposures” (cve.mitre.org) deu a essa falha o código CAN-2005-0399.

Uma falha foi encontrada na maneira como o Mozilla exibe caixas de diálogo. Em situações em que diversas páginas estejam abertas em abas, uma página maliciosa que esteja numa aba não ativa (ou seja, que não esteja sendo mostrada) poderia apresentar ao usuário uma caixa de diálogo, levando-o a acreditar que essa caixa pertence à página ativa (ou seja, à página que o usuário está lendo no momento). O projeto “Common Vulnerabilities and Exposures” (cve.mitre.org) deu a essa falha o código CAN-2004-1380.

Há ainda outra falha, desta vez na maneira como o Mozilla autoriza os plugins a carregar conteúdo privilegiado em um *frame* HTML (com a tag `<frame>`). A falha permite que uma página insidiosa possa persuadir o usuário a clicar em certos lugares, o que poderia provocar a reconfiguração do sistema ou mesmo a execução de código arbitrário. O projeto “Common Vulnerabilities and Exposures” (cve.mitre.org) deu a essa falha o código CAN-2005-0232.

Não é só o navegador que possui problemas. Um comportamento anormal no *Mozilla Mail*, mais precisamente nas rotinas que tratam os famigerados *cookies*, pode causar dores de cabeça. Ao receber os cookies quando carrega conteúdo sobre o protocolo HTTP – por exemplo, um email em HTML com um *frame* que carrega uma página externa –, o Mozilla Mail tende a não respeitar as configurações do usuário quanto à forma de manipulá-los. É possível rastrear um usuário com o uso de mensagens de email maliciosas que carregam conteúdo por HTTP. O projeto “Common Vulnerabilities and Exposures” (cve.mitre.org) deu a essa falha o código CAN-2005-0149.

O Mozilla faz das suas também na forma como responde a solicitações para autenticação em servidores de proxy. Um servidor web malicioso poderia roubar credenciais

do navegador da vítima se emitir uma resposta do tipo 407 (*proxy authentication request*, ou *solicitação de autenticação via proxy*). O projeto “Common Vulnerabilities and Exposures” (cve.mitre.org) deu a essa falha o código CAN-2005-0147.

As rotinas de validação tags HTML também tem problemas. Algumas tags de início, se seguidas pelo caracter NULL, podem tirar o navegador de sua órbita. Uma página maliciosamente preparada poderia travar o Mozilla assim que a vítima tentasse abri-la. O projeto “Common Vulnerabilities and Exposures” (cve.mitre.org) deu a essa falha o código CAN-2004-1613.

Uma falha foi encontrada na forma como o Mozilla lida com as permissões para a instalação de pacotes XPI – a maioria das extensões e temas, bem como alguns dos plugins do Mozilla, são distribuídos nesse formato. É possível que um pacote XPI deixe alguns dos arquivos com permissão de leitura (ou escrita) para

todos os usuários do computador, permitindo que qualquer usuário local possa roubar dados sigilosos ou executar código arbitrário. O projeto “Common Vulnerabilities and Exposures” (cve.mitre.org) deu a essa falha o código CAN-2004-0906.

Uma falha foi encontrada na forma como o Mozilla abre novas abas cada vez que se “clica” com o botão do meio em algum link. Uma página maliciosa poderia ler arquivos locais ou reconfigurar o *chrome*. O projeto “Common Vulnerabilities and Exposures” (cve.mitre.org) deu a essa falha o código CAN-2005-0141.

E a última do mês para o Mozilla. Uma falha foi encontrada na maneira como o programa mostra o ícone indicador de site seguro. Um site malicioso poderia carregar, em segundo plano, uma página realmente segura enquanto exibe na tela uma página com truques para enganar, de alguma forma, o usuário. Como está carregando “por trás” uma página segura, o Mozilla irá

mostrar o ícone de site seguro, embora a página sendo exibida não o seja. O projeto “Common Vulnerabilities and Exposures” (cve.mitre.org) deu a essa falha o código CAN-2005-0144. ■

Referência no Gentoo: GLSA 200503-30 / Mozilla

Referência no Red Hat: RHSA-2005:323-10

gtk2

O pacote *gtk2* contém o famoso *GIMP Toolkit* (GTK+), uma biblioteca com funções para criar interfaces gráficas que funcionem sob o X Window System.

Uma falha foi encontrada na maneira como o GTK 2 processa imagens BMP. É possível usar imagens BMP especialmente manipuladas para causar uma negação de serviço no aplicativo que usa essa biblioteca. O projeto “Common Vulnerabilities and Exposures” (cve.mitre.org) deu a essa falha o código CAN-2005-0891. ■

Referência no Mandriva: MDKSA-2005:068

Referência no Red Hat: RHSA-2005:344-03z

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/glsa/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.

Notícias do kernel

❑ A novela do BitKeeper...

...terminou, afinal. Linus Torvalds parou de usar o sistema de controle de versões *BitKeeper*. Como decorrência disso, temos observado o esforço da comunidade de desenvolvedores para encontrar um substituto à altura da função – esforço esse fascinante de se observar. A decisão foi repentina e tanto Linus como os outros desenvolvedores estão ocupados vasculhando os quatro cantos em busca de uma maneira de continuar o desenvolvimento do kernel da maneira como estavam acostumados – ou seja, com um volume de código estonteante. E é precisamente esse alto volume de trabalho que Larry McVoy, CEO da BitMover, disse ser impossível de ser acomodado por qualquer das ferramentas de controle de versão desenvolvidas pela comunidade, pelo menos a médio prazo. Esperemos para ver se o homem que fabrica o BitKeeper tem razão ou se é apenas papo. Nesse caso, provavelmente um software desenvolvido pela comunidade tomará seu lugar.

Obviamente, é bastante provável que Linus e seus comparsas encontrem (ou mesmo desenvolvam) algo que funcione para eles, da mesma forma como o venerável sistema de envio de *patches* funcionou nos velhos tempos – atendia ao Linux, mas não podia ser aplicado como solução de uso geral para outras comunidades de desenvolvedores. Com isso, uma das equipes com maior volume de trabalho, hoje, é justamente a que desenvolve o conjunto de scripts 'temporários' para que o kernel possa continuar sendo desenvolvido enquanto Linus procura por

uma solução permanente. O pai de todos já avisou: está pensando até mesmo em adotar o *Subversion* como solução paliativa caso algo concreto não seja encontrado nos próximos dias, embora tanto ele como os desenvolvedores do Subversion concordem que essa não é a melhor ferramenta para uso com o kernel Linux. Se isso acontecer, é porque a coisa degingolou mesmo – mas é bastante possível e nem tão improvável assim.

De qualquer forma, um número razoável de aplicativos promissores estão sendo preparados para preencher a lacuna. Embora o *arch* não seja consenso, em grande parte devido a seu intratável mantenedor, há um grande número de projetos que são ou desdobramentos (*forks*) diretos do *arch* ou reimplementações de seus recursos básicos. Um deles é o *bazaar-ng*, de autoria de Martin Pool, um “mano sangue bom” no desenvolvimento do kernel. Ele vem trabalhando como um alucinado para tornar seu rebento suficientemente usável para os desenvolvedores do kernel – pelo menos como solução ‘interina’.

Outro forte candidato é o *monotone*, um sistema de controle de versões distribuído que Linus já tinha elogiado antes, classificando-o de “a melhor das alternativas livres”. Mas embora o monotone seja um sistema indubitavelmente ótimo, ainda precisa de bastante suor para ser minimamente útil para as necessidades dos desenvolvedores do kernel. Não ficou claro, ainda, se a maciça atenção que o projeto recebeu recentemente está ajudando ou atrapalhando o esforço frenético de sua equipe de desenvolvimento.

Uma estranha – e fascinante – abordagem do problema está no sistema de arquivos *git*, desenvolvido por... Linus Torvalds! Ele já está funcionando em termos aceitáveis, mas isso não deve ser interpretado como “já está pronto pro povo lá fora”. Também não deve ser inferido que Linus quer criar um sistema de controle de versões no sentido tradicional da expressão. O que quer que esteja saindo da mente alucinada de Torvalds, será algo novo. Ele sugeriu, entretanto, que um sistema tradicional de controle de versões poderia ser construído usando o *git* como alicerce – mas ele próprio não moverá uma palha nessa direção; apenas implementará os recursos que julgar importantes. Ao mesmo tempo, Linus espera largar o fardo do desenvolvimento do *git* sobre costas alheias, mantendo as rédeas de mantenedor e reservando-se o direito de aceitar (ou recusar) patches. ■

❑ CPUs Geode finalmente no kernel

Kianusch Sayah Karadji tem mantido há bastante tempo um patch que dá ao kernel 2.6 suporte às CPUs Geode. O patch é essencialmente independente e estava, até então, disponível apenas em seu site pessoal. Mas como cada vez mais usuários escreviam pedindo que fosse incorporado ao kernel oficial, Kianush finalmente decidiu enviá-lo para inclusão na árvore principal. O patch deu entrada no final de fevereiro – tarde demais para entrar na versão 2.6.11, lançada no dia 2 de março – mas só apareceu na 2.6.12-rc1 algumas semanas depois. Se nada

de anormal for identificado no código, vai aparecer definitivamente na versão estável 2.6.12 assim que for lançada. Kianusch tem ajudado os desenvolvedores do kernel com scripts desde 2002, mas essa é sua primeira contribuição ao kernel propriamente dito. ■

❑ Hotplug Redux

Greg Kroah-Hartman tirou da cachola (e do papel) um subsistema de *hotplug* que roda no espaço do usuário. Chamado de *hotplug-ng*, a intenção é substituir o sistema atual, baseado em scripts do *Bash*, por um binário compilado de tamanho diminuto. Greg reimplementou a coisa em C para evitar a necessidade de se ter o interpretador *Bash* instalado, bem como para reduzir as necessidades de espaço em disco e memória de todo o sistema. A velocidade conseguida, entretanto, é de longe o maior benefício alcançado pela reimplementação. Em testes empíricos e sem qualquer medição precisa, os scripts em *Bash* levaram de dois a três segundos para fazer os dispositivos funcionarem, enquanto o binário compilado levou menos de um segundo. Nem o próprio Greg tinha notado a relevância de seu feito e só percebeu que era algo realmente formidável quando o pessoal dos sistemas embarcados começou a cantar e dançar de alegria. Os desenvolvedores do SUSE LINUX, por exemplo, detectaram esperas de até 40 segundos que podem ser reduzidas para dois ou três se a reimplementação de Greg for usada. ■

❑ Cartões de memória SD

Pierre Ossman mostrou a seus pares seus esforços para tornar os cartões de memória do tipo *Secure Digital* (SD) compatíveis com Linux. Esses cartões podem armazenar vários gigabytes e se propõem a facilitar o transporte de arquivos multimídia entre uma grande variedade de tipos de hardware. Esses cartões também possuem um sistema de

controle de direitos autorais e propriedade intelectual: há restrições para os tipos de dados que podem ser extraídos do cartão. Alguns desenvolvedores do kernel se mostraram empolgados por ver algo sendo produzido nesse terreno, mas infelizmente Pierre não teve acesso a informações técnicas suficientes para terminar o trabalho e, em especial, para ter seu código incluído no kernel oficial. Isso se deve em parte a questões jurídicas: será que há permissão legal para escrever drivers para os cartões SD? ■

❑ Novo subsistema central timeofday

John Stultz vem trabalhando num subsistema de gerenciamento de hora no sistema que, para começar, é independente de plataforma. Com isso ele espera reduzir a cruel complexidade e a infame duplicidade (ou, antes dizer, multiplicidade) de código que torna essa área do kernel uma verdadeira esfinge para os desenvolvedores. Mas tomar uma parte do código que aparentemente depende da plataforma de hardware e torná-la genérica o bastante para funcionar em qualquer lugar é um desafio digno dos personagens de Júlio Verne. Algumas das arquiteturas têm porções desse código no espaço do usuário, enquanto outras implementam tudo no kernel. Além disso, qualquer código que John venha a apresentar deve ter um desempenho igual ou superior ao código que ele pretende substituir. Para aplacar o sofrimento que toda mudança acarreta, John criou um “gancho” opcional para o novo subsistema – cada arquitetura fará uso desse *hook* quando mais lhe aprouver. ■

❑ SysFS mostra os primeiros sinais de fadiga

Uma tentativa recente de corrigir a localização de um diretório mal-posicionado no *SysFS* deu com os burros n’água: um número não desprezível de outros subsis-

temas já esperam que o diretório exista naquela posição. Esse é, precisamente, o tipo de situação que fez do sistema de arquivos */proc* o pesadelo dos desenvolvedores e usuários. O conjunto de mantenedores do *SysFS* tentou ao máximo evitar as armadilhas (e o destino) de seu antecessor *ProcFS*, mas esse incidente inexoravelmente indica que os mesmos problemas ocorrerão no futuro.

Os desenvolvedores do *SysFS* estão a par do histórico desastroso do *ProcFS*, dos *ioctl*s e do diretório */dev*: todos os três viraram terríveis “chupa-cabras”. Em vista deste passado horripilante, é de se esperar que o *SysFS* tenha vida muito mais longa que seus ancestrais. Com um pouco de otimismo podemos dizer que, se o “capenga” *ProcFS* pôde ser usado por décadas, talvez o *SysFS*, mais robusto, esteja em plena vida útil daqui a alguns séculos.

O grande problema encarado por todos esses sistemas (incluindo o *SysFS*) é a maneira como uma interface consistente (ou seja, que faça um mínimo de sentido) deve ser apresentada ao usuário. O paradoxo é que essa interface tem de se manter simples e, ao mesmo tempo, ser compatível com um número cada vez maior de serviços, número esse que cresce exponencialmente. A quantidade de hardware a ser suportado cresce sem limites – e todos eles precisam ser representados no */dev* de uma forma que os usuários comuns possam reconhecer e usar. Os drivers e outros pontos do kernel crescem, mudam, são divididos em categorias diferentes (e, por vezes, dividem uma mesma categoria com drivers que, no passado, estavam em outras) e, ainda assim, devem aparecer de uma forma consistente, útil e à qual o usuário já esteja acostumado. O problema existe desde tempos imemoriais, quando o Unix ainda não era escrito na linguagem C, e continuará a existir até que a última variante de Unix seja extinta. ■



Wilson Godoy - Assessoria de Imprensa Microsiga

Entrevista com Wilson Godoy

Gestão brasileira com Linux

Entre as empresas que atuam na área de pacotes de gestão empresarial, os ERPs, a Microsiga merece destaque por seu crescimento em diversos segmentos da indústria, comércio, serviços, governo, saúde e outros, contando ainda com soluções de *Business Intelligence*, gestão de recursos humanos e *Supply Chain*. Com atuação também em outros países da América Latina, a empresa recentemente se fundiu com a Logocenter, resultando num grupo que faturou R\$ 380 milhões no ano passado – entre seus clientes estão 56% de todas as pequenas e médias empresas do Brasil. Essa cifra deve chegar a R\$ 420 milhões neste ano. O Linux também está presente nos negócios da empresa: dos seus 8 mil clientes no Brasil, 20% já migraram partes ou todo seu sistema de gestão para a plataforma Linux. A Linux Magazine conversou com Wilson Godoy, vice-presidente de tecnologia e sistemas.

POR ALEXANDRE BARBOSA

Linux Magazine» Qual é a história do Linux dentro da Microsiga?

Wilson Godoy» Quando começamos a aprimorar nossa plataforma própria de software, de 1995 a 1996, já sentíamos a necessidade de portar o nosso sistema, o Proteus, para outras plataformas que não o Windows®. O primeiro ambiente, evidente, foi o da Microsoft, até por uma questão de demanda do mercado; isso foi em agosto de 1999. Mas nós já tínhamos uma base de conhecimento em Unix muito sólida e, mesmo antes de lançar o sistema, já tínhamos um plano de desenvolvimento para Linux, tanto que lançamos uma versão do sistema para a plataforma poucos meses depois, em março de 2000. Havia a percepção de uma tendência de crescimento fora do Brasil e, como tínhamos interesse na internacionalização da solução, fizemos uma aposta. Pessoalmente, eu achava que o Linux avançaria muito rápido nas pequenas empresas no curto prazo. Bem, eu errei. Isso aconteceu mais nas médias empresas, porque elas estavam mais estruturadas para bancar as migrações e por ter estrutura para contar com ou contratar pessoal especializado para fazer a manutenção em Linux, coisa que não acontece na pequena empresa.

LM» O sistema teve uma adoção rápida?

WG» Nem tanto. Em 2000, a base de clientes era pouco significativa e isso não mudou muito nos dois anos seguintes. Acho que em 2002 nossa base de clientes em Linux não chegava a 1%. Mas isso também era reflexo do amadurecimento da plataforma e da estrutura de serviços no Brasil. Tanto é que, hoje, entre 15% e 20% da plataforma servidor de nossos clientes está em Linux. Hoje temos até alguns clientes usando o Linux no front-end da operação, o que nos levou também a desenvolver e lançar versões de nosso software para operação em desktops com Linux.

LM» Como foi o desenvolvimento dessa versão cliente do software?

WG» Começamos a trabalhar essa versão direto em Linux e acabamos usando o *toolkit* Qt, da Trolltech, o mesmo que usamos até hoje. No início criamos uma interface em C, mas recomecemos do zero com um único cliente que roda nas duas plataformas.

LM» Em sua opinião, há uma distribuição Linux que mereça destaque?

WG» Posso dizer que estamos mais próximos do SUSE LINUX.

LM» *Quantos desenvolvedores a empresa tem? Vocês usam o Linux no desenvolvimento de novos produtos?*

WG» Contamos com uma equipe de 250 desenvolvedores. Em termos de desenvolvimento propriamente dito, usamos uma plataforma desenvolvida internamente, a ADVPL. Foi uma escolha que fizemos depois de apanhar um bocado ao usar ferramentas de mercado e tendo que sofrer na hora de mudar de uma linguagem para outra. Mas meus programadores, no dia-a-dia, têm a liberdade de escolher que plataforma operacional ou ferramentas vão usar e o Linux está bem presente na nossa equipe.

LM» *Como você vê o crescimento do Linux no ambiente corporativo?*

WG» Nos grandes centros o Linux é uma realidade. A plataforma já decolou. Tanto é que eu não conheço nenhuma organização que não conte com pelo menos um servidor rodando algum tipo de aplicação em Linux. O mesmo já não acontece quando você avança para pólos regionais. Acredito que isso aconteça por falta de mão-de-obra nessas regiões, ausência de profissionais ou empresas capazes de prestar serviços de qualidade e que possam dar suporte a aplicações, a projetos de implantação ou de migração, gente que possa dar sustentação a uma mudança para o Linux.

LM» *Como o uso do Linux afeta as operações da Microsiga?*

WG» Temos clientes que migram para o Linux porque estão em processo de atualização de máquinas e porque querem baixar os custos com as licenças de servidor, que é a razão principal que motiva os clientes a pensar no Linux. Estabilidade é um atributo no qual mesmo o ambiente Windows fez progressos significativos, se você comparar as versões atuais com o que rodava cinco anos atrás. O curioso é que mesmo o cliente que não adquire a

nossa solução em Linux se sente confortável ao saber que a empresa fornece essa opção pois, se decidir mudar de ambiente amanhã, sabe que poderemos ajudá-lo na transição. E, como temos experiência nisso, podemos trazer mais segurança para empresas. Lembre-se de que o ERP é parte vital dos negócios das empresas e que processos vitais como o faturamento não podem ser interrompidos. Já tive casos de clientes com 150 usuários que dormiram com Windows® na sexta-feira e acordaram com Linux na segunda seguinte. Outra coisa é a garantia de independência em relação ao fornecedor. Não é o caso, mas se amanhã sofrermos algum tipo de pressão hostil de fornecedores de sistemas operacionais, sabemos que sempre haverá uma alternativa extensível aos nossos clientes. Mas falar de ambiente operacional é tranquilo, porque está na base da pirâmide de um sistema computacional. Entretanto, quanto mais elevada a posição da aplicação na escala de criticidade da empresa, mais a lógica do software livre vai precisar de ajustes.

LM» *Ajustes de que natureza?*

WG» Existem partes de uma solução que podem ser críticas para a empresa ou podem contar diferenciais para a operação. No caso do ambiente operacional, não há razão para não usar o Linux. Indo mais acima na escala de funcionalidade, chegamos aos bancos de dados. Existem organizações que ficarão à vontade com BDs Open Source. Outras não. Quando você chega ao nível da aplicação, o modelo Open Source pede uma forma de administração mais segura. Na Microsiga nós fizemos esse ajuste, mais especificamente com o sistema Prefeitura Livre. Trata-se de uma versão voltada para governo eletrônico cujo código fonte está sendo aberto à colaboração de desenvolvedores do setor público. Mas, para que não se perca a rastreabilidade e a qualidade do software, todas as mudanças são submeti-

das a análise. É preciso ter em mente que estamos lidando com um ERP, ou seja, qualquer mudança vai afetar o funcionamento do sistema como um todo. Dessa forma, deixamos o programador pesquisar novos caminhos, ao mesmo tempo em que criamos uma contingência para evitar que alterações acidentais causem danos aos sistemas. Eu sei que isso foge um pouco do praticado pela comunidade do Software Livre.

LM» Mas é mais ou menos o que o próprio Linus Torvalds faz...

WG» Sim. Nosso objetivo não é restringir a inovação, mas garantir que sistemas complexos ou delicados serão manipulados por pessoas com a devida bagagem. E trazer mais segurança ao processo é algo que está de acordo com as preocupações do governo, que é o maior incentivador atual do progresso do Software Livre no Brasil.

LM» Que bancos de dados vocês usam em sua solução?

WG» Utilizamos as principais soluções de mercado, como Oracle, DB2, Informix, Sybase e CTree, além do SQL Server. Em Software Livre, nossos clientes podem usar MySQL ou PostgreSQL.

LM» Já existe uma base significativa de clientes que utilizam os bancos de dados de código aberto?

WG» Ainda são poucos. As médias empresas precisam de segurança, o que se traduz num suporte sempre presente, então elas buscam soluções de mercado nas quais vão encontrar o apoio devido na forma de serviços. São poucas as opções de suporte de nível corporativo em atividade com os bancos de dados de código aberto.

LM» Há algum perfil definido do cliente Microsiga que opta pelo Linux?

WG» Nada desse tipo. Temos hospitais, empresas de transportes... não há um padrão. Existem, sim, alguns centros

regionais. Goiânia, por exemplo, é um centro regional forte em Linux. Lá temos menos clientes em Linux do que em São Paulo, mas o percentual de empresas usando este sistema operacional é maior que em todo o resto do país.

LM» Na hora da venda, a Microsiga explora o Linux como argumento de venda? Na comparação com o Windows, o quanto a versão Linux sai mais barata para o cliente?

WG» Somos bem isentos em termos de plataforma. Procuramos respeitar as escolhas do cliente e mostrar que temos opções e que a solução será segura independentemente do caminho decidido por ele. Não há diferença, a rigor, entre as versões para Windows® e Linux. O cliente vai economizar nas licenças de servidores e pode baixar custos com bancos de dados, uma vez que, na aquisição do nosso software, há uma licença embutida do DB2, da IBM.

LM» A Microsiga atua também em outros países. O comportamento do mercado, quanto ao Linux e Software Livre, é diferente nesses países?

WG» É muito semelhante ao Brasil. Se pegarmos os mercados em que estamos mais ativos, como México, Chile e Argentina, estamos mais ou menos no mesmo estágio. Temos clientes com Linux em todos esses países. Mas não é de se estranhar. Veja: em países nos quais a cultura de Unix foi mais forte – como aconteceu na Europa – a transição para o Linux é mais fácil. Isso não aconteceu na América Latina. No Brasil, junto com a reserva de mercado, tivemos o problema de migrar diretamente dos minicomputadores rodando sistemas proprietários para os microcomputadores; daí a explosão e rápido crescimento do Windows® e do Netware, sem abrir muito espaço para criar uma cultura de Unix que alimentasse o Linux atualmente.

LM» Que obstáculos você vê para o progresso do Linux?

WG» Acho que o caminho para o crescimento do Linux está no desenvolvimento de aplicações. No lado servidor, as plataformas existentes estão bem maduras, mas na hora em que você vai desenvolver software corporativo, por exemplo, é preciso contar com ferramentas de modelagem e de depuração de código. Ao chegar ao mercado, o que a empresa de software descobre? Que as versões para Linux dos aplicativos dessa natureza podem custar de seis a dez vezes mais que as mesmas ferramentas em versão para Windows®. Há alguma razão que justifique isso? Então um dos entraves é o alto custo de desenvolvimento para os produtores de aplicativos.

LM» Quais são as tendências tecnológicas vinculadas ao Linux?

WG» Temos o regime de ASP (*Application Service Provider*). Não é uma inovação, mas é uma tendência. Atualmente, para o cliente Microsiga com até 15 usuários, fica mais barato usar a ferramenta via Web do que manter um sistema internamente. Além disso, os custos de conexões ADSL são hoje muito menores do que eram há alguns anos, além do que, ao compartilhar recursos, o custo por usuário cai drasticamente. É uma saída interessante para os pequenos, porque a grande corporação faz o outsourcing, enquanto a pequena empresa precisa do ASP. E nosso serviço se apóia na dupla DB2-Linux para viabilizar os custos mais baixos. Temos cerca de quatro ou cinco novos clientes nessa modalidade, que já conta com mais de 1200 usuários em 80 contratos. Há ainda a virtualização de máquinas. Um cliente nosso tinha um mainframe e decidiu continuar com ele, mudando o sistema interno e criando várias partições, gerando máquinas virtuais. Hoje temos nosso sistema em duas dessas partições. ■

A praça é do povo!

No início dos tempos, a web era uma via de mão única. Alguma entidade paranormal se esgueirava por detrás dos servidores, estalava os dedos e, *presto!* O conteúdo aparecia automaticamente, esculpido com cuidado para o deleite dos internautas – mas ainda estático, perene, imutável. Depois disso, a divindade mandou ao universo virtual anjos que eram não apenas arautos de seu santo senhor mas que tinham permissão também para conversar conosco, reles mortais – eram os formulários em HTML e os scripts em CGI, ferramentas que trouxeram os usuários para mais perto de seus deuses. Mesmo assim, “falar” com os donos do site ainda era algo extremamente controlado, cuidadosamente orquestrado pelo trabalho árduo de desenvolvedores web muitíssimo bem pagos.

Mas os internautas queriam mais: falar com essas mitológicas criaturas, os donos da informação, já não era mais satisfatório. O próximo *round* da reinvenção da Internet foi marcado por uma reviravolta nas comunicações entre internautas. Se antes eles passivamente liam o conteúdo provido por pessoas ou entidades que sequer conheciam, hoje os usuários querem ser, também eles, deuses da informação e divulgar, para quem queira conhecer, suas próprias idéias e opiniões. O tema de nossa revista deste mês é a popularização da Internet e da fácil publicação de conteúdo – por qualquer pessoa – por meio de ferramentas que facilitam a vida do usuário não-técnico: o *Wiki* e o *Blog*.

Essas ferramentas levam o conceito de cooperação e colaboração na Web a níveis jamais vistos. E o melhor de tudo: ninguém precisa de doutorado em engenharia aeroespacial para usá-las. Mostraremos como um usuário comum de Linux pode configurar rápida e facilmente um site colaborativo usando as mesmas ferramentas livres e de código aberto que os especialistas amam. Prepare-se, pois veremos coisas pra lá de bacanas, como por exemplo:

⇒ **Mediawiki** – a ferramenta por trás da famosa e controversa enciclopédia colaborativa *Wikipedia*. O *Mediawiki* foi idealizado para organizar os esforços conjuntos de comunidades virtuais. Os usuários podem inserir e editar conteúdo, notas e revisar todos os documentos – tudo via Web. O *Mediawiki* possui um sistema de controle de versões, controle de acesso e muitas outras funções úteis para um trabalho cooperativo eficiente e simplificado.

⇒ **TWiki** – Assim como o *Mediawiki*, o *TWiki* é um Wiki de código aberto. Você aprenderá como instalar e configurar o programa, bem como ficar a par de todos os meandros de sua interface com o usuário. Daremos ênfase especial aos desafios da implementação e uso de um sistema Wiki em ambiente empresarial e mostraremos como usar os melhores recursos do *TWiki* em seu próprio ambiente comunitário.

⇒ **Drupal** – um verdadeiro “feiticeiro” que o ajuda a construir sites. Mostraremos como usar o *Drupal* para criar comunidades, blogs e sistemas web personalizados.

⇒ **Zope X3.0** – O *Zope* é um framework para sistemas de gerenciamento de conteúdo muito popular escrito em Python. Mostramos as principais novidades da versão X3.0.

Seja você um profissional da Internet, um voluntário em uma comunidade ou um usuário regular de Linux, temos certeza de que encontrará algo de útil no seu dia-a-dia e muita diversão nesta edição sobre Wikis e Blogs. Afinal, “a praça pertence ao povo”.

Esforço comunitário 22

Conheça o *MediaWiki*, o sistema por trás da popular enciclopédia on-line *Wikipedia*.

Construtor de equipes 28

TWiki, um excelente software para construção de sistemas colaborativos para o mercado empresarial.

Conteúdo a conta-gotas 35

Mostramos com detalhes todos os passos da instalação e configuração do *Drupal*.

Fórmula X 42

Veja o que há de novo no *Zope X3.0*.

Configurando e administrando o MediaWiki 1.4

Esforço comunitário

Os fãs dos wikis são mal-acostumados. Não raro, encontram um erro de digitação em uma página que estão visitando e, via de regra, saem clicando pela tela toda na vã esperança de encontrar um botão de editar. Até a criação do princípio wiki, era impensável que os próprios visitantes pudessem editar um site – ainda mais de uma pessoa que não conhecessem ou de uma empresa à qual não estão ligados. Hoje tudo mudou: projetos como a enciclopédia livre *Wikipedia* [1] claramente demonstram a viabilidade do esforço colaborativo, que pode envolver milhares de voluntários. Muitas empresas e projetos de código aberto já adotaram wikis em áreas como a documentação de software ou a organização de documentos em uma intranet empresarial.

Se você está interessado em criar seu próprio wiki, ficará admirado com a oferta de soluções disponível. Mais de 50 mecanismos de wiki foram desenvolvidos nos últimos anos usando uma grande variedade de linguagens de programação. O *MediaWiki* [2], sistema usado pela *Wikipedia*, é uma ferramenta que provou seu valor nos últimos dois anos, tendo ficado em estado permanente de uso sob tráfego intenso. Esse teste de fogo fez com que os desenvolvedores do *MediaWiki* descobris-

sem falhas de segurança e as corrigissem rapidamente. Resultado: o *MediaWiki* é um dos mecanismos de wiki mais estáveis, seguros e testados do mundo.

Instalação

Para funcionar, o *MediaWiki* precisa de um servidor web Apache com o módulo PHP instalado e o banco de dados MySQL. Os scripts de administração precisam do interpretador PHP para a linha de comando, que os usuários do Debian vão encontrar num pacote à parte chamado *php4-cli*. Para criar miniaturas de imagens (*thumbnails*), precisamos ainda do programa *ImageMagick* ou da biblioteca gráfica do PHP, a GD. O Apache deve ser configurado para permitir a execução externa de código PHP.

Baixe a versão mais atual do *MediaWiki* do site oficial [2]. Neste artigo usaremos a versão *1.4beta5*, mas a versão 1.4 final deve estar “nas prateleiras” no momento em que esta revista chegar às bancas.

Como root, abra um console no servidor e mude para o diretório onde baixou o *MediaWiki*. De lá, copie o arquivo recém-baixado para o diretório raiz de documentos (*document root*) do Apache – no Debian, por exemplo, ele é */var/www* – e descompacte o arquivo:

A melhor maneira de organizar um esforço colaborativo na web é através de um wiki. O *MediaWiki*, o mecanismo por trás da famosa *Wikipedia*, dita os padrões para todas as outras ferramentas de colaboração via web.

Por ELISABETH BAUER

```
cp mediawiki-1.4beta5.tar.gz /var/www
cd /var/www
tar xzvf mediawiki-1.4beta5.tar.gz
```

Um novo diretório será criado; o nome desse diretório será parte da URL na qual seu wiki será acessado (por exemplo, se o diretório chamar-se *jurubeba*, o wiki será acessado pela URL <http://www.seusite.com.br/jurubeba>). Se você quiser mudar o nome do diretório e colocar algo mais curto (*wiki*, por exemplo) digite:

```
mv mediawiki-1.4beta5 wiki
```

Agora mude para o diretório do wiki e altere as permissões de acesso do diretório de configuração:

```
cd wiki
chmod a+w config
```

Depois de completar a instalação, não se esqueça de tornar o diretório apenas de leitura novamente. Agora, rode o script de instalação dentro de seu navegador. Basta acessar o endereço: <http://localhost/wiki/config/index.php>.

O script verifica os requisitos para instalação. Se vir uma mensagem de erro nesse ponto, em vez da caixa de

diálogo mostrada na [figura 1](#), volte algumas etapas e corrija as configurações do Apache e do PHP. O script também verifica a presença do ImageMagick ou da extensão GD do PHP para que possamos gerar miniaturas de imagens.

Informe o nome de seu wiki no campo *Site name*:. Esse nome aparecerá em muitas posições de seu wiki como o nome do projeto e o MediaWiki criará um espaço na tela chamado de *project name* com esse título. Não entendeu? Não se preocupe, voltaremos a isso mais tarde. Por ora, apenas escolha e informe um nome para seu wiki.

Também é necessário um endereço de email para que o sistema envie mensagens de erro. Depois, use o controle de idiomas para selecionar que língua seu wiki vai falar. Além do idioma, é necessário informar a codificação de caracteres. Selecionar *us - English - Unicode* permite adicionar itens ao wiki usando caracteres acentuados sem que seja necessário fazer uso de entidades HTML (como *à* e *´*; etc.). Existe a opção da língua portuguesa, mas como ela não está codificada em UTF-8 você poderá ter problemas de acentuação aqui e ali – nada que assuste, entretanto. Como a tradução é bastante incompleta, talvez você fique melhor com a versão em inglês mesmo.

Em uma intranet ou rede privada, essa coisa de licenciamento do conteúdo não é lá muito importante. Se o wiki será acessível ao público em geral, entretanto, é melhor pensar a respeito. Uma das opções oferecidas pelo script de instalação é a *GNU Free Documentation License* (licença de documentação livre GNU), inspirado na GPL e aplicável a textos e outros documentos. Essa licença permite a livre reprodução, modificação e uso comercial do conteúdo, mas garante o reconhecimento da autoria do material. Já a licença *Creative Commons* permite mais controle sobre a informação. Por exemplo, é possível proibir o uso com-

ercial do material armazenado e escolher um texto de licença apropriado a seu país (há inclusive uma versão adequada ao Brasil). Clique em *choose* (escolher) para ir ao site da Creative Commons e escolher a versão da licença que mais lhe apetece. O MediaWiki aceita sua escolha e mostra um link para o texto da licença no rodapé de todas as páginas.

É preciso especificar uma conta para o usuário que irá administrar o wiki. O nome padrão é *Sysop*, mas você pode escolher um outro. O *Sysop* (*System Operator*, nome herdado das antigas BBSs) terá privilégios de divindade sobre o wiki.

Para melhorar o desempenho do MediaWiki é possível fazer “cache”, ou seja, armazenar cópias pré-processadas, do conteúdo mais acessado. O instalador oferece a possibilidade de usar tanto o *Turck MMCache* como o *Memcached* – mas você só pode escolher uma dessas opções se tiver instalado os caches antes. O Debian possui pacotes prontos para ambos. O *Turck MMCache* é um cache escrito (e pseudo-compilado) em PHP e é muito útil para

rodar o MediaWiki em um único servidor. A Wikipedia trabalha com uma “fazenda” de servidores e usa o *Memcached*, um sistema de cache distribuído originalmente desenvolvido para o *Livejournal* e que roda como um *daemon*.

Na etapa final, cuidamos do banco de dados. O script de instalação cria um usuário e um banco de dados no MySQL. É possível usar os padrões na maioria dos casos. Se você quer que o MediaWiki use o mesmo banco de dados que outros sistemas (situação muito comum nos provedores de hospedagem brasileiros) informe o nome do banco e um prefixo para as tabelas do MediaWiki (*Database table prefix*). Se estiver indeciso, use *mw_*. Isso evitará que o MediaWiki escreva em uma tabela que já exista no seu banco de dados.

Digite a senha do administrador da base de dados (root) no último campo, *DB root password*. Esta **não é** a senha de *root* do Linux, mas sim a do MySQL. Elas podem ser iguais ou diferentes, e algumas distribuições sequer definem uma senha por padrão. ➔

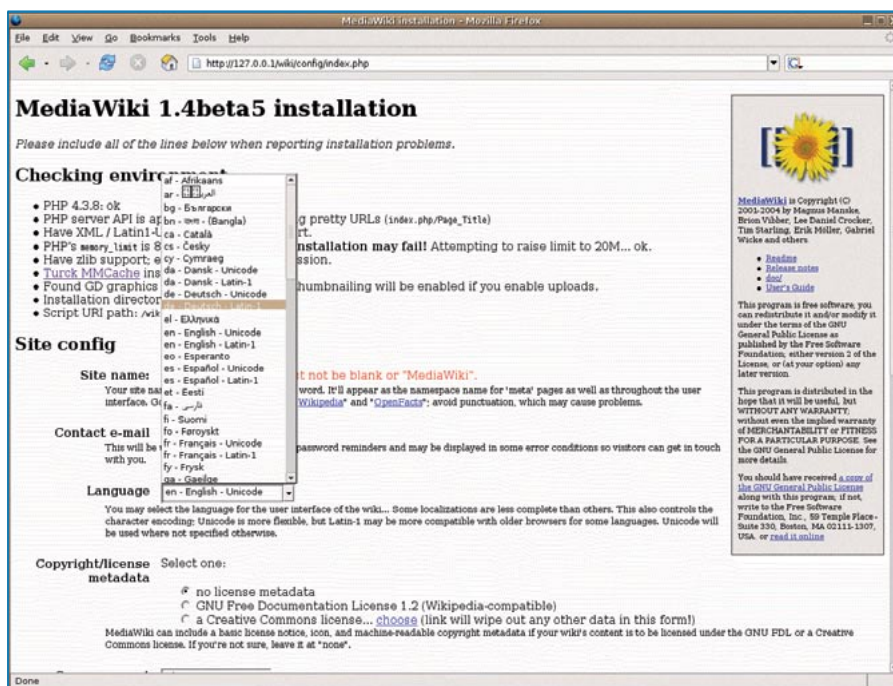


Figura 1: Um script PHP chamado de dentro de um navegador é tudo o que você precisa para instalar o MediaWiki.

Agora clique em *Install*. O script criará o banco de dados (ou usará um já existente), um usuário para o MySQL e um para o MediaWiki, as tabelas no banco, povoará a tabela com informações de exemplo e gravará tudo num arquivo chamado `LocalSettings.php` no diretório `config`. Se tudo funcionar como esperado, surgirá uma mensagem enorme berrando “Sucesso!”, uma explicação do que fazer com o arquivo `LocalSettings.php` e um link para a página principal do wiki.

Nesse ponto você deve fazer o que está escrito. Antes de visitar o wiki, é preciso tomar o primeiro bonde em direção ao console e mover o arquivo `LocalSettings.php` para o diretório principal do wiki. Abra um terminal, vá até a pasta principal do wiki e digite:

```
mv config/LocalSettings.php .
```

Pronto! Clique no link e você será apresentado com a página inicial do wiki (ver [figura 2](#)).

Primeiros passos

Clicar na aba *edit*, presente na “barra de ferramentas” no topo da página do wiki, leva à janela de edição. Nela, os visitantes podem encher a página com conteúdo. Recursos especiais podem ser obtidos usando a sintaxe wiki (ver [tabela 1](#)). O conteúdo é em texto simples e possui umas poucas marcas (*tags*) de formatação. Digitar * no início de uma linha cria um marcador de lista, por exemplo. Sinais de igual podem ser usados para controlar o tamanho dos títulos e subtítulos. Podem ser usados conjuntos de um, dois, três ou quatro sinais, por exemplo:

```
==Heading==
```

Uma linha em branco cria um parágrafo novo. Mas os links são talvez o aspecto mais fascinante do MediaWiki. Ao contrário dos wikis clássicos como o *Usemo-*

dwiki e o *TWiki* (veja matéria sobre ele na página [28](#) desta edição), que se baseiam em um esquema de detecção automática de links calcada no formato *CamelCase*, o MediaWiki permite links escritos em texto puro. Por exemplo, links para páginas internas são definidos dentro de pares duplos de `[[colchetes]]`. Digite:

```
[[Página de Teste]]
```

e grave as alterações clicando no botão *Save page* (salvar página). O MediaWiki mostrará, na posição em que a palavra foi inserida, um link na cor vermelha cujo nome é [Página de Teste](#). Se clicarmos nesse link, outro editor se abrirá e você pode editar essa nova página – que, obviamente, não existe ainda. Para voltar à página principal, crie um novo link com o nome dessa página. Em inglês, será `[[Mainpage]]`; em português, `[[Página principal]]`. Esse link será mostrado em azul, já que a página já existe. O manual do MediaWiki possui mais informações sobre a criação de páginas (veja [\[3\]](#)).

Como o MediaWiki foi desenvolvido com o objetivo de ser um mecanismo para enciclopédias, diverge dos wikis tradicionais em alguns pontos. Por exemplo: o MediaWiki introduz o conceito de espaços de nomes (*namespaces*), prefixos imutáveis que aparecem na frente do título da página. Há ainda uma página de discussão para qualquer assunto tratado no wiki, com uma aba na “barra de ferramentas” que leva a ela. Cada um dos usuários possui uma página pessoal, que pode ser referenciada por outras páginas com um link do tipo `[[User:nome]]`. Com isso, os usuários podem externar opiniões, fazer um tipo de blog, tomar notas para seu próprio uso ou simplesmente apresentar-se aos outros usuários. Se alguém deixa uma mensagem na página de discussões pessoais do usuário, o MediaWiki informa o fato a ele na próxima vez que entrar em sua página pessoal. O software inclui uma página descritiva com um prefixo *image* para imagens e os usuários podem editá-la como qualquer outra página no wiki.

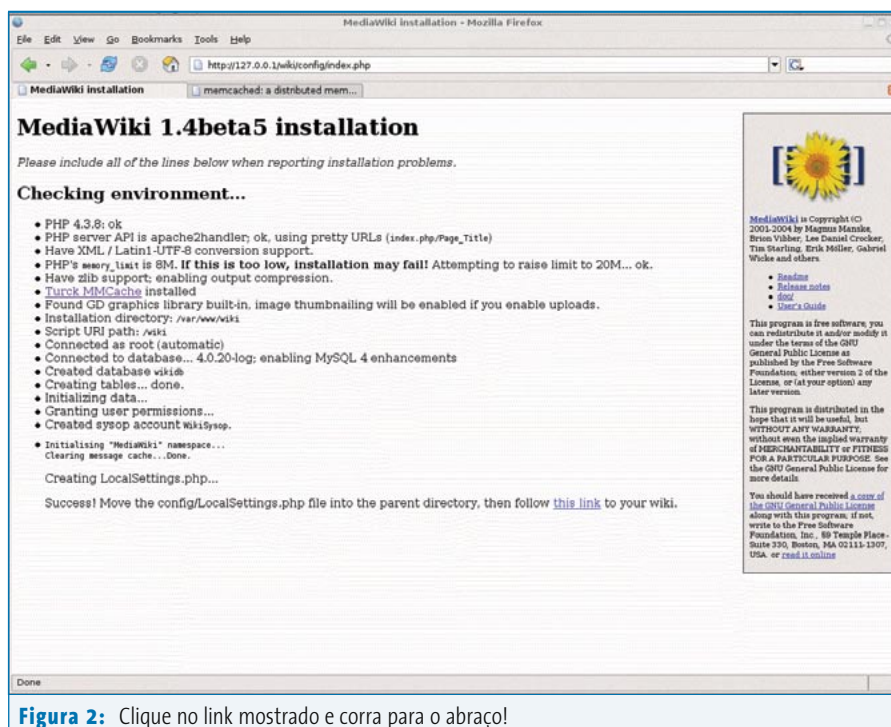


Figura 2: Clique no link mostrado e corra para o abraço!

Os usuários podem criar e editar páginas no espaço de nomes *Template* (em português seria *modelo*, mas a versão portuguesa do MediaWiki traduziu isso como *predefinição*). Uma marcação como `{{Melancia}}` permite criar uma página predefinida que pode ser usada como molde ou modelo para outras páginas. O modelo é guardado num endereço especial. Em nosso exemplo, seria <http://localhost/wiki/index.php/Template:Melancia> (ou [Predefinição:Melancia](#) se estiver usando a versão em português).

Configurações

A instalação padrão parece boa o bastante para fazer bonito na maioria das situações. Entretanto, é óbvio que cedo ou tarde você pode querer personalizar seu wiki – seja substituindo a imagem da flor por um logotipo só seu, seja adicionando um texto de boas-vindas ou substituindo o tema padrão do wiki. É possível fazer uma boa parte dessas personalizações diretamente no wiki, desde que tenha iniciado uma sessão de uso com a conta do sysop. As outras configurações são verdadeiramente manuais: é preciso abrir seu editor de textos favorito e editar a unha o arquivo `LocalSettings.php`.

As opções padrão podem ser encontradas em `includes/DefaultSettings.php`. Se quiser mudar alguma delas, copie os valores necessários para `LocalSettings.php` e as altere ali, pois o arquivo `DefaultSettings.php` costuma ser substituído sem aviso quando o software é atualizado.

Para substituir o logotipo, informe o caminho para ele em `LocalSettings.php`:

```
$wgLogo = "/caminho/para/seu/logotipo/2meulogo.png";
```

Para modificar quaisquer dos textos mostrados na interface do usuário, acesse a página <http://localhost/wiki/index.php/Special:Allmessages> (em português: [Especial:](#)

[Allmessages](#)). A página que surge mostra uma lista com todas as páginas do espaço de nomes *MediaWiki*, que em vez de guardar conteúdo guarda as mensagens mostradas, itens dos menus e títulos das páginas. Cada página corresponde a uma mensagem do sistema, como por exemplo *Login successful* (em português, *Agora você está logado como...*).

O MediaWiki 1.4 e posteriores permitem que os usuários modifiquem a interface em sua própria língua. O programa importa as mensagens da linguagem padrão apenas na fase de instalação, jogando-as no espaço de nomes *MediaWiki*. Para todos os outros arquivos, o sistema obtém as mensagens do diretório *languages*. Mas é possível modificar as mensagens mostradas diretamente no wiki. Por exemplo, `MediaWiki:Mensagem/xx` apresenta a tal mensagem na língua indicada por `xx`, que é o código ISO do idioma desejado (como *pt*, *em*, *es*, *ja*, etc...).

As mensagens do MediaWiki permitem que se altere, inclusive, os links no menu à esquerda. A URL <http://localhost/wiki/index.php/MediaWiki:Currentevents> permite redirecionar o link *Current events* (Eventos atuais) para uma página mais apropriada em seu wiki. Você pode usar marcas como `MediaWiki:Portal` e `MediaWiki:Portal-url` para associar um novo título e endereço a seu projeto. Para evitar que usuários maldosos possam ficar alterando a interface o tempo todo, esses recursos são restritos a usuários com privilégios administrativos.

Em vez de criar um tema a partir do nada, também é possível usar o espaço de nomes *MediaWiki* para alterar a aparência da página. Com `[[MediaWiki:Monobook.css]]` e `[[MediaWiki:Monobook.js]]` pode-se modificar muitos dos elementos do tema padrão, chamado *Monobook*, usando CSS e Javascript. As páginas `[[User:Nome/Monobook.css]]` e `[[User:Nome/Monobook.js]]` permitem que os usuários alterem suas próprias páginas pessoais.

Imagens

Com a exceção do recurso de envio de imagens, seu wiki já está praticamente completo e funcional. Mas por motivos de segurança, o subsistema de envio de arquivos está desativado. Para ativá-lo, retire o sinal de comentário (`#`) da linha `# $wgDisableUploads = false;` no arquivo de configuração do MediaWiki. O Apache precisa de permissão de escrita no diretório *images*. A melhor maneira de garantir isso é usando o comando *chown* para atribuir a posse do diretório ao usuário apropriado – que no caso do Debian é o *www-data*.

Certamente você gostaria de impedir que código PHP fosse executado no diretório de imagens ou mesmo que arquivos HTML não pudessem ser interpretados pelo navegador. Isso impediria tentativas de invasão por meio de vulnerabilidades do PHP e do HTML. Configure o Apache para evitar esses dissabores:

```
<Directory "/var/www/wiki/images/">
    php_admin_flag engine off AddType 2
    text/plain .html .htm .shtml .php
</Directory>
```

Apenas usuários registrados podem enviar imagens ao wiki. Para se certificar de que isso realmente funciona, insira um link *Create account* (inscrever conta) na parte superior direita da tela. Assim os usuários que desejarem poderão criar suas próprias contas de acesso. Após o usuário fazer *login* o menu à esquerda deve, agora, exibir um formulário para envio de arquivos.

O MediaWiki reconhece apenas alguns tipos de arquivos. Se quiser estender essa lista, copie todos os valores de `includes/DefaultSettings.php` para `LocalSettings.php` e adicione aí os novos tipos. Veja o exemplo a seguir:

```
$wgFileExtensions = array( 'png', 'gif', 2
'jpg', 'jpeg', 'ogg' );
```


Se o wiki estiver em uma rede local, atrás de um firewall e com usuários confiáveis, a verificação de tipos pode ser desligada:

```
$wgCheckFileExtensions = false;
```

Os arquivos são adicionados às páginas da mesma forma que as imagens: `[[Image:xuxubeleza]]` adicionará à página uma imagem chamada `xuxubeleza.png`. Se o ImageMagick ou a biblioteca GD estiverem instaladas, você pode até incluir uma miniatura da imagem com a tag `[[Image:xuxubeleza|thumb|30px|descrição]]`. Clicar na miniatura leva a uma página com a imagem e sua descrição. As versões do MediaWiki superiores à 1.4 permitem montar galerias de fotos (`<gallery>`):

```
<gallery>
Image:kdegnome|Descrição: turma do KDE e 2
do Gnome em uma briga de rua.
Image:Pordosol.png
Image:Cabrita.gif
</gallery>
```

O MediaWiki registra todos os uploads de arquivos em um relatório (figura 3).

É possível adicionar links para arquivos que o navegador não pode renderizar diretamente – por exemplo, `[[Media:ManualGentoo.pdf]]` ordena ao Media

Wiki que crie um link de download para um suposto manual do Gentoo Linux em formato PDF.

Controle de acesso

Os wikis nasceram calcados em duas idéias revolucionárias: edição simplificada de conteúdo e liberdade para que qualquer um modificasse suas páginas – qualquer um mesmo, não precisa “fazer login” pra isso. Muitos dos wikis modernos como a Wikipedia, a paródia alemã *Kamelopedia* [4] (onde nada deve ser levado a sério) ou o guia de viagens *WikiTravel* [5] ainda ostentam essa bandeira.

Esses wikis abertos são mantidos por uma comunidade de voluntários que verificam periodicamente todos os assuntos e removem rapidamente as ocorrências de spam, vandalismo e falta de bom senso. O controle de versões integrado e as listas de monitoração permitem que todos os usuários acompanhem as alterações dos assuntos de seu interesse. Isso torna o MediaWiki ideal para os operadores desses wikis abertos.

Se você é um sujeito desconfiado até da sua mãe, ou se prefere evitar ter que todo dia limpar a “caca” dos outros, é possível promover alguns dos usuários (escolhidos a dedo) a administradores. No extravagante mundo da administração

de permissões do MediaWiki, apenas os burocratas mais aguerridos conseguirão usar a página `[[Special:Makesysop]]`. A conta criada durante a instalação já tem essa opção ativada.

Usuários com contas administrativas podem remover páginas, protegê-las contra edição por usuários normais e bloquear vândalos notórios em uma página especial `[[Special:Blockip]]`. Nessa página, é possível configurar o afastamento do indivíduo

associando números às palavras *days*, *hours* e *weeks* (em português, *dias*, *horas* e *semanas*). A palavra chave *infinite* bane o agitador até o fim de sua vida.

Por padrão, o MediaWiki só permite que os administradores bloqueiem endereços IP. Para bloquear um usuário em especial ou faixas de IP, adicione as seguintes linhas:

```
$wgSysopUserBans = true;
$wgSysopRangeBans = true;
```

ao arquivo `LocalSettings.php`. Depois do email e dos blogs, a nova fronteira do spam são os wikis. Se você pretende oferecer um wiki ao público em geral, deve verificar todas as suas páginas diariamente à cata de spam, a fim de remover o material ofensivo e livrar a vista de seus usuários de propaganda não-solicitada.

Para prevenir o chamado *linkspam*, que serve para promover os endereços dos spammers em mecanismos de busca, os desenvolvedores introduziram a opção `$wgNoFollowLinks` no MediaWiki 1.4. Quando ativada (que é, por acaso, a condição de fábrica), ela ordena ao MediaWiki que inclua o atributo `rel="nofollow"` a todos os links externos, evitando que os robôs de spam possam detectá-los. Obviamente, isso afeta também os links legítimos – problema resolvido com a opção `$wgNoFollowLinks = false;`.

Outra alternativa é usar expressões regulares para filtrar spam – o MediaWiki se negará a aceitar conexões que combinem com o padrão da expressão configurada em `LocalSettings.php`.

```
$wgSpamRegex = "/((http:\\\\(www\\2
\\.)(spammer1.com|spammer2.vu)/)";
```

A utilidade desse método é limitada. Os administradores precisam atualizar o arquivo de configuração depois de cada incidente; a lista das URLs bloqueadas pode crescer expo-

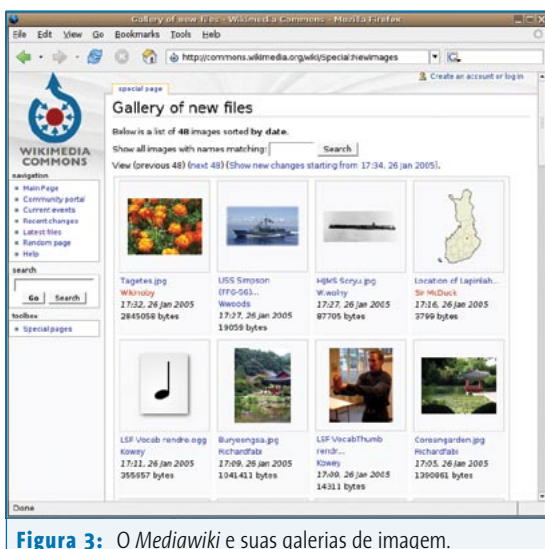


Figura 3: O Mediawiki e suas galerias de imagem.

Tabela 1: Sintaxe Wiki

==Título 1==	Título, primeiro nível
===Título 2===	Título, segundo nível
====Título 3====	Título, terceiro nível
'texto'	Qualquer texto entre dois pares de aspas simples fica <i>italico</i>
'''texto'''	Qualquer texto entre aspas simples e duplas aninhadas fica negrito
* Item	Lista não numerada (os * são substituídos por marcadores)
# Item	Lista numerada (os # são substituídos por números sequenciais)
-	Linha horizontal
[[Título da página]]	Link para uma página dentro do wiki
[[Título Outro texto]]	Link para uma página dentro do wiki; em vez do nome da página é mostrado <i>Outro texto</i>
http://exemplo.com	URL externa (automaticamente convertida em um link)
[http://exemplo.com Outro texto]	URL externa (automaticamente convertida em um link) em vez da URL é mostrado <i>Outro texto</i>
[[Image:File.jpg]]	Adiciona uma imagem à página
[[Image:File.jpg thumb]]	Adiciona automaticamente um thumbnail (miniatura) para uma imagem existente
{{Xubiruba}}	Integra dinamicamente o conteúdo atual no modelo <i>Template:Xubiruba</i> (ou <i>Predefinição:Xubiruba</i>)
{{subst:EnderecoCartas}}	Insere o conteúdo de <i>Template:EnderecoCartas</i> (ou <i>Predefinição:EnderecoCartas</i>) na página atual
{{:Maricota}}	Integra dinamicamente o conteúdo atual na página <i>Maricota</i>
~~~~	Cria um campo de assinatura para o usuário registrado – em outras palavras, um link para a página pessoal do usuário dentro do wiki
~~~~~	A mesma assinatura, mas com data e hora.

nencialmente. Que tal, então, ativar a extensão experimental SpamBlacklist Extension [6], que a Wikipedia vem usando desde dezembro? Baixe o arquivo disponível no repositório CVS do MediaWiki, hospedado no SourceForge. Com ele, os administradores podem gerenciar as “listas negras” (*blacklists* – listas contendo os endereços consagrados de spammers contumazes) como se fossem páginas dentro do *wiki*. Para que os spammers não editem a página, removendo seus próprios endereços, deixe-a trancada e com acesso liberado apenas para um dos administradores.

Todas essas medidas de segurança são aplicáveis a *wikis* públicos. Se você não tem tempo de ficar verificando as coisas e mexendo nas páginas o tempo todo, restrinja as permissões de leitura, escrita e criação de usuários no MediaWiki. Assim,

só o pessoal autorizado poderá alterar o site. Para isso, adicione as linhas a seguir ao arquivo `LocalSettings.php`:

```
$wgWhitelistEdit = true;
$wgWhitelistRead = array ( "Special:2
Userlogin", "Mainpage");
$wgWhitelistAccount = array ( 'user' => 2
0, 'sysop' => 1, 'developer' => 1 );
```

A primeira linha restringe o acesso a usuários registrados, a segunda especifica quais páginas os usuários não-registrados podem visitar (pelo menos uma página é obrigatória: a de login. Você pode adicionar mais se quiser). A matriz `$wgWhitelistAccount` permite especificar os usuários autorizados a criar novas contas – neste exemplo, os usuários precisam de *status* de administrador ou de desenvolvedor. As três declarações podem ser aplicadas separadamente.

Extensões

O MediaWiki agora inclui uma fabulosa coleção de extensões e utilitários externos, que pode ser encontrada em [8]. Dentre essas ferramentas, desenvolvidas para facilitar o trabalho com enciclopédias, incluem-se linhas de tempo, hieróglifos, editor de fórmulas matemáticas (em sintaxe LaTeX), uma extensão para notação musical (usando o *Lilypond*) e muitas outras coisas. Se você não tem tempo para se manter informado das atualizações em seu *wiki*, confira o *Enotif* de Thomas Gries [9]. O programa (na realidade, um *patch*) adiciona novas opções de configuração ao MediaWiki, permitindo que os usuários sejam notificados por email quando alguma página do site for modificada.

Muitos administradores evitam ao máximo tarefas que consumam enormes quantidades de tempo – por exemplo, modificar os links para uma longa lista de páginas ou reescrever texto. O *PyBot* [10] é uma estrutura para a construção de robôs (*bots*) escrito em Python. Criado para o MediaWiki, ele permite automatizar um grande número de tarefas.

INFORMAÇÕES

- [1] A enciclopédia pública Wikipedia (em inglês - substitua *en* por *pt* para ter a versão em português): en.wikipedia.org
- [2] MediaWiki: mediawiki.org
- [3] Manual do MediaWiki: tinyurl.com/c2cp5
- [4] Kamelopedia, uma paródia da Wikipedia (em alemão e inglês): kamelopedia.de
- [5] Guia de viagens Wikitravel: wikitravel.org
- [6] Listas negras de SPAM: tinyurl.com/7964u e tinyurl.com/duxol
- [7] Folhas de estilo: tinyurl.com/9ne28
- [8] Ferramentas e extensões do MediaWiki: tinyurl.com/9oq6n e tinyurl.com/bvod9
- [9] Enotif: tinyurl.com/76tug
- [10] Pybot: sourceforge.net/projects/pywikipediaibot

Wikis no mundo corporativo

Construtor de equipes

Um wiki pode manter unidos os grupos de trabalho dentro das empresas, organizando as coisas, poupando tempo, adiantando cronogramas e permitindo a colaboração entre todos. O TWiki é tanto um software colaborativo de código aberto para o mundo empresarial como uma plataforma para construir soluções web.

POR PETER THOENY

Os wikis estão “pegando”. Este certamente será o ano do wiki da mesma forma como o ano passado foi a vez dos blogs. Os wikis são ferramentas que permitem a construção de um site colaborativo, no qual todos os usuários e visitantes podem incluir informação, editá-la, organizá-la e apresentá-la de forma coerente para os outros visitantes. São, portanto, muito úteis no ambiente empresarial, pois as equipes podem gerar documentação, planos de negócio, relatórios e afins sem precisar se reunir formalmente: a informação é inserida e manipulada se e quando estiver disponível. O wiki mais famoso do universo é a *Wikipedia* [1], a enciclopédia online em que os próprios usuários podem adicionar, editar e corrigir verbetes. Além da Wikipedia, muitos sites que adotam o paradigma wiki habitam a Internet, como o original *WikiWikiWeb* [2] de Ward Cunningham.

Mas pode um software “anárquico” como o wiki ser usado em ambiente corporativo? À primeira vista, um wiki é algo bastante desestruturado e por demais caótico para ser usado no dia-a-dia da maioria das empresas – afinal, os executivos normalmente querem “tudo certinho dentro do vidrinho” e não são muito afeitos a nada que não possam controlar. Há também o aparente risco de segurança e uma falsa sensação de que nada pode ser rastreado. Para surpresa de muitos – fato demonstrado, aliás, por aqueles que adotaram precocemente a tecnologia – um wiki funciona às mil maravilhas se guardado por detrás de um firewall.

Intranets mantidas por *webmasters* são notória e vergonhosamente desatualizadas. O conteúdo é, tipicamente, incompleto e encontrá-lo é, via de regra, um desafio. Inconsistências de informação pululam entre os departamentos. Ferramentas especiais, conhecimento e permissão de acesso são necessários para manter esse conteúdo. A razão para todos esses problemas é a chamada “síndrome do webmaster”: se um usuário desco-

bre uma página com conteúdo incorreto ou incompleto, normalmente ignora o fato, pois encontrar quem é o webmaster responsável por ela e redigir um email com a correção dá muito trabalho. Pior ainda: os poucos que fazem isso tendem a ser solenemente desdenhados por webmasters relapsos.

O surgimento dos wikis mudou tudo isso. O conteúdo não é mais mantido por um distante e antipático webmaster, mas por quem realmente tem conhecimento para gerá-lo. Uma fábrica de parafusos, por exemplo, pode ter as especificações e projetos de seus produtos inseridos na intranet pelos engenheiros mecânicos e não pelo webmaster, que afinal não entende nada de parafuso. Ao mover parte ou todo o conteúdo da intranet para o wiki, os visitantes podem ler as informações disponíveis como se estivessem num site comum; entretanto, esses mesmos visitantes são imbuídos do poder de adicionar dados novos, editar dados inconsistentes ou incorretos e corrigir problemas na hora em que são identificados. Evita-se assim o antigo e ineficaz processo de identificar uma falha, solicitar a correção e esperar meses para que algo seja feito. Não há software adicional a ser instalado nas máquinas dos usuários, já que tudo pode ser feito por uma interface bastante intuitiva em seus próprios navegadores – os investimentos com treinamento e suporte ficam próximos do zero. Diga isso a seu gerente. Ele vai gostar.

Os wikis também têm o efeito de corrigir os mal-entendidos causados por emails. Uma mensagem de correio eletrônico usa o sistema “um escreve, o outro responde”: você só pode adicionar coisas à primeira mensagem, mas não consegue mudar o que já foi dito. O paradigma do email é maravilhoso para a fase de discussões, mas para chegar “aos finais” e redigir um documento final é um pesadelo – já notou que, na sua empresa, discussões iniciadas por

email acabam em uma reunião presencial com todos os envolvidos e muito papel impresso? Outra ineficiência do email: não é estruturado e não pode ser ligado a outros documentos – trocando em miúdos, a informação não pode ser agrupada com assuntos afins de forma fácil. Os anexos não possuem controle de versão, dificultando o rastreamento do histórico do documento. Um wiki pode auxiliar sobremaneira a vencer todas essas ineficiências do sistema de correio virtual. Em vez de enviar um email detalhado da primeira vez, basta mandar um email simples dizendo “olha, o problema está rabiscado no wiki neste endereço”. Os outros integrantes da equipe podem, então, continuar a montar a versão final do documento diretamente no wiki, que registra quem disse o quê e mantém todas as versões do documento num banco de dados.

Um wiki é, também, flexível o suficiente para ser a base de sistemas inteiros baseados na web. O formato wiki é ideal para criar aplicativos estruturados com formulários, consultas e relatórios talhados para os funcionários que irão usar a ferramenta.

Os wikis empresariais têm uma necessidade adicional. Antes de implementar um wiki em sua intranet, os administradores da rede corporativa precisam ter certeza de que a ferramenta escolhida seja de fácil entendimento, que o setor de suporte tenha facilidade para ajudar o usuário a usá-la, de que ela tenha a segurança necessária e possa ser integrada com toda a parafernália de TI já existente na empresa.

Muitos wikis fizeram história no ambiente corporativo (veja o quadro **as empresas e seus wikis**). Neste artigo nos concentraremos no TWiki, um wiki de código aberto desenvolvido especialmente para o ambiente empresarial e que se presta muito bem como alicerce de sistemas corporativos bastante complexos.

Mas, então, o que é o TWiki?

O TWiki é um enorme projeto de código aberto hospedado no site twiki.org [3]. Muitos desenvolvedores contribuem com idéias, especificações técnicas, código e documentação. A missão do TWiki é ser uma plataforma de colaboração calcada nas tecnologias da web mais avançada do mundo. Seu público alvo são as intranets corporativas, onde o TWiki se propõe a promover o fluxo de informações dentro da organização, permitir que equipes distribuídas trabalhem em uníssono de forma transparente e produtiva e acabar com a “síndrome do webmaster”, praga horrenda das intranets desatualizadas. O TWiki pode ser usado como:

- Bloco de rascunho coletivo para os membros de um projeto;
- Ferramenta de colaboração entre departamentos da empresa;
- Sistema de administração de intranets;
- Uma base de conhecimento digital;
- Uma plataforma para a criação de aplicações baseadas na tecnologia web, como portais, sites de notícias,

Quadro 1: As empresas e seus wikis

Wikis de código aberto especialmente desenvolvidos para corporações de qualquer tamanho existem às pencas. Aqui está uma pequena lista deles (entre parênteses, temos a linguagem na qual o software foi desenvolvido):

- *PHPWiki*: uma implementação cheia de recursos e que trabalha com uma grande variedade de bancos de dados (PHP);
- *Tiki*: um CMS com módulos para Wiki, fóruns à la Slashdot, blogs, galerias de imagens, salas de bate-papo e outros badulaques (PHP);
- *TWiki*: uma das mais entupidas de recursos, pode ser usada como fundação para aplicações empresariais e possui centenas de plugins (Perl);
- *XWiki*: outra implementação bem completa, que pode usar inclusive alguns plugins do TWiki (Java);
- *ZWiki*: Um wiki que funciona na plataforma Zope (Python).

sistemas de inventário, de rastreamento de falhas... As possibilidades são praticamente ilimitadas.

O TWiki normalmente é instalado na rede interna, atrás de um firewall. Muitas empresas grandes usam o TWiki como sistemas de administração de grupos (groupware) por ser muito mais amigável ao usuário do que sistemas comerciais como o Lotus Notes. Para ter uma idéia de como as empresas estão usando wikis para melhorar a comunicação de suas equipes, leia os casos de sucesso do TWiki [4], escritos por pessoas associadas a corporações gigantescas como a Disney, Motorola, SAP e Yahoo!

Instalando o TWiki

Vá até a página oficial do TWiki, em twiki.org [5] e baixe a última versão estável do programa. Descompacte o arquivo em um diretório temporário em seu servidor. O pacote deverá conter os seguintes diretórios:

- **bin** : scripts em Perl para ser usados como CGIs;
- **lib** : bibliotecas do Perl;
- **pub** : espaço para armazenar arquivos enviados pelos usuários;
- **data** : dados das páginas e registros de eventos (*logs*);
- **templates** : Modelos e temas (*skins*) para os sites e para a interface de administração do TWiki.

O processo de instalação do TWiki é manual. Siga o tutorial (que é bem completo, por sinal) encontrado no arquivo *WikiDocumentation.html*, localizado no diretório raiz do pacote. Um administrador experiente consegue instalar o TWiki em menos de uma hora, incluindo aí todo o trabalho de configuração do Apache, das permissões de acesso apropriadas para todos os arquivos, do sistema de autenticação e do daemon *cron* – este último usado para enviar notificação de alterações nas páginas aos usuários interessados.

O TWiki na berlinda

A melhor maneira de aprender sobre o TWiki é explorar sua interface. O ambiente contém *webs* (grupos) e *topics* (assuntos). Uma web é como se fosse um wiki dentro do TWiki. Cada web é um vasilhame cheio de assuntos. Um assunto é uma página que pode ser manipulada (editada) pelos usuários. A figura 1 mostra o assunto *GnuSkinGoButton* localizado na web *Support* do site twiki.org. Observe que os nomes da web e do assunto são refletidos na URL. Em nosso exemplo, a URL é <http://twiki.org/cgi-bin/view/Support/GnuSkinGoButton>. Os números na figura 1 referem-se aos recursos do TWiki que iremos abordar neste artigo.

As webs do TWiki (item 1 da figura 1) estão referenciadas na barra de navegação no lado esquerdo do assunto. Cada departamento ou grupo pode ter sua própria web com seu próprio conjunto de nomes. A web principal (*main*) e a chamada *twiki* são especiais. A web *main* é a web primária e contém todos os usuários registrados e seus grupos. Já a web *twiki* abriga a página de administração e a documentação.

Os membros da equipe podem ver o que mudou recentemente em uma web simplesmente clicando no link *Changes* (item 2 da figura 1). Também é possível se inscrever em uma web alheia, mais precisamente no assunto *WebNotify*. Com isso, o usuário será avisado diariamente por email se alguma mudança ocorrer em algum dos assuntos dessa web. Há também transmissões de notícias pelo protocolo RSS [6] para quem prefere essa maneira de ser informado. Todos os três métodos permitem que os membros da equipe fiquem atualizados e possam revisar, corrigir e aprimorar o conteúdo constantemente.

O TWiki possui um mecanismo de busca que procura por assuntos na web atual (item 3 da figura 1). Os usuários mais “nerds” podem refinar a busca com expres-

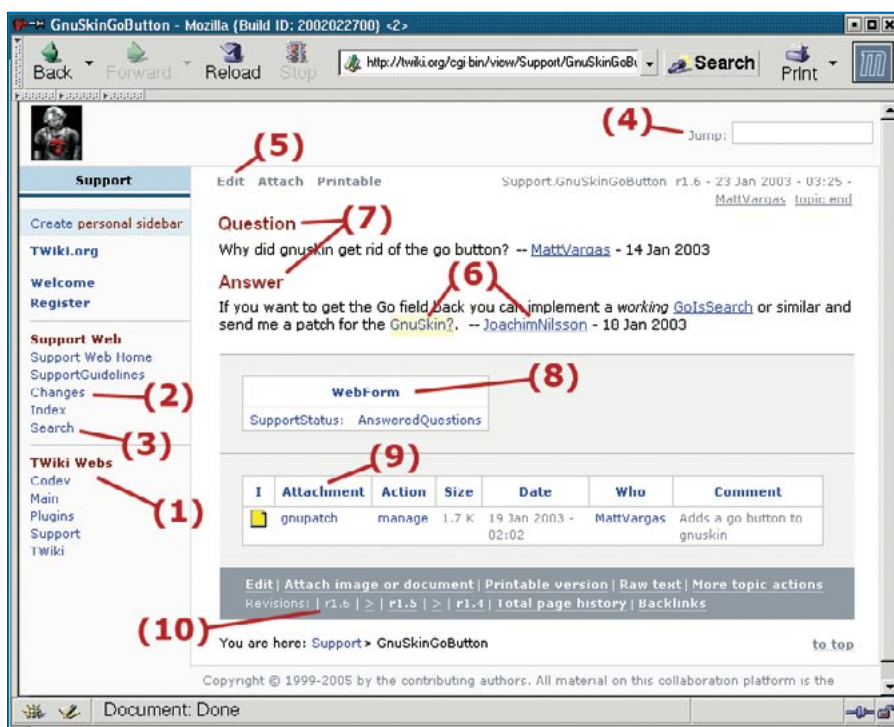


Figura 1: O ambiente interativo do TWiki com (1) webs, (2) alterações, (3) busca, (4) navegação, (5) edição, (6) links, (7) temas, (8) formulários, (9) arquivos anexados e (10) revisões.

sões regulares. O TWiki também faz uso de variáveis [7] para compor dinamicamente o conteúdo de certas páginas. Por exemplo, uma busca pode ser “enlatada” e embutida em um assunto com a variável `%SEARCH{...}%`. Para aplicações web, isso é o máximo! O parâmetro de busca pode ser passado como uma URL usando a variável `%URLPARAM{"search"}%`.

A caixa de navegação (item 4 na [figura 1](#)) permite pular de um lado a outro no ambiente virtual do TWiki. Se você quiser ser teletransportado ao assunto “inscrição”, por exemplo, escreva *WebNotify* no campo apropriado e pressione **[Enter]**. Se você sabe que existe um assunto denominado “objetivos” mas não lembra o nome exato, digite a palavra *objetivos* para obter uma lista que possua essa palavra em seu nome. Para pular para uma outra web, acrescente um ponto final . ao termo informado. Por exemplo, para ser transferido para a web “Marketing”, digite *Marketing.* (não esqueça o ponto). Obviamente, deve existir uma web com esse nome.

Como em qualquer wiki, a função mais importante é a de edição (item 5 da [figura 1](#)). Esse link convida todo mundo a contribuir com o assunto. O TWiki possui um sistema de marcas fácil de aprender (ver [figura 2](#)). Escrever uma página no TWiki é como redigir um email – apenas escreva, sem se preocupar com a formatação. Dica: o caracter de sublinhado () indica que o texto está em *itálico*, não sublinhado, como seria de se esperar. Há um editor de HTML saindo do forno [9] e que deve ser integrado ao TWiki nas próximas versões. Os assuntos são travados durante uma sessão de edição. Se mais alguém tentar editar um tópico enquanto outro usuário está trabalhando nele, receberá um alerta a respeito.

Fazer interligações entre assuntos é um recurso-chave para colaboração eficiente entre as equipes (item 6 da [figura 1](#)). Nosso cérebro funciona baseado em associações

e precisamos apenas ouvir, vez por outra, uma determinada palavra para recuperar todo um contêiner de informação associado a ela – pense em sua mãe e você se lembrará de como toda a sua adolescência foi infeliz. Um wiki trabalha de forma parecida. As pessoas são convidadas a criar, a seu bel-prazer, ligações entre assuntos. Para isso, usamos as chamadas *WikiWords* – ou palavras em *CamelCase*, como queira. As *WikiWords* são palavras capitalizadas e escritas sem espaços entre elas (por exemplo, *RelatorioConsolidado* ou *FerramentasDeIntranet*). Para criar um link para um assunto que você já sabe que existe, digite sua *WikiWord*, como mostra o exemplo a seguir:

Para mais informações, consulte **?**
FerramentasDeIntranet.

Quando o assunto é salvo, a *WikiWord* se transforma automaticamente num link:

Para mais informações, consulte **?**
FerramentasDeIntranet.

Observe que, ao contrário do HTML, não é preciso especificar uma âncora para o link. O TWiki faz isso automaticamente quando reconhece uma *WikiWord*. E o que acontece se eu digitar uma *WikiWord* mas ela nunca foi usada (*NeverUsedWord*)? Em outras palavras, não existe um assunto com o nome digitado. O sistema reconhece a *WikiWord* mas, como o assunto não existe, convida o usuário a criar um – isso é indicado pela interrogação ao lado da palavra. Para criar esse tópico, clique na interrogação, digite o texto e salve-o. Se voltarmos ao assunto original, veremos que a *NeverUsedWord* desapareceu e tornou-se uma *WikiWord* propriamente dita.

O TWiki pode ser embelezado com temas (item 7 da [figura 1](#)). Quando um novo assunto é criado, o usuário vê um texto padrão (tipicamente uma tela em

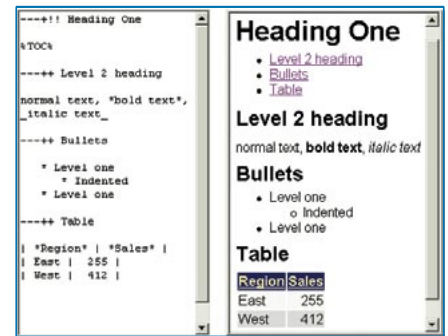


Figura 2: Exemplo das marcas no TWiki.

branco com uma assinatura datada no rodapé). Isso é definido pelo tema global em uso, que pode ser configurado em *WebTopicEditTemplate*, localizado na web TWiki. Para que uma web em especial use um tema exclusivo, crie um assunto de mesmo nome em *WebTopicEditTemplate*. Uma aplicação montada a partir do TWiki que use formulários precisa que seus assuntos sejam criados baseando-se num tema personalizado. Isso pode ser feito com um formulário HTML embutido em um assunto do TWiki que possua uma marca de entrada chamada *templatetopic*, que por sua vez insira o nome do tema para esse assunto [10].

Adicionando um método de entrada baseado em formulários a um conteúdo mais livre (item 8 da [figura 1](#)) podemos criar assuntos estruturados com ilimitadas categorias e, com isso, desenvolver um aplicativo baseado no fluxo de trabalho da equipe. Quando os formulários são ativados em uma dada web e atribuídos a um assunto, um formulário aparece quando entramos em modo de edição. A tela fica dividida em duas grandes áreas: a de cima é o conteúdo “desorganizado” que estamos acostuma-

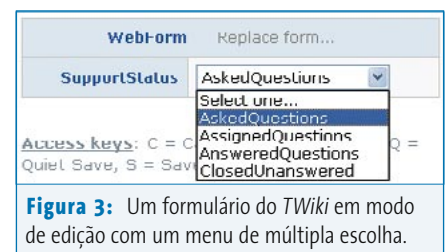


Figura 3: Um formulário do TWiki em modo de edição com um menu de múltipla escolha.

dos a ter nos wikis, enquanto a parte de baixo possui um formulário com tudo o que tem direito: botões, caixas de texto, menus, caixas de seleção e botões de rádio. O que quer que esteja nesse formulário será mostrado como uma tabela quando a página estiver sendo exibida a um visitante.

Por exemplo, o rastreador de falhas (ou bugs) disponível no site twiki.org usa um formulário TWiki para organizar o fluxo de trabalho desta forma:

- ➔ 1. Alguém preenche um formulário na web *Support*, fazendo uma pergunta (veja referência [11]).
- ➔ 2. O aplicativo cria um novo assunto com o conteúdo do formulário. O assunto também tem um formulário com um campo chamado *SupportStatus* com o valor *AskedQuestions*. Isso indica que essa é uma pergunta ainda não respondida.
- ➔ 3. Todos os assuntos com o estado *AskedQuestions* são listados na página inicial da web *Support*. Isso é feito com a marcação `%SEARCH{}`.
- ➔ 4. Alguém no suporte técnico está sem fazer nada e decide parar de jogar paciência para trabalhar um pouquinho. Escolhe uma das perguntas sem resposta e a edita. Na parte inferior do formulário, o atendente altera o estado de *AskedQuestions* (ver [figura 3](#)) para *AssignedQuestions*, além de escrever algum conteúdo no assunto.
- ➔ 5. Se voltarmos à página inicial do suporte, essa pergunta está agora listada na coluna *Assigned* (atribuído).
- ➔ 6. Quando o atendente souber resolver o problema do usuário, altera o estado para *AnsweredQuestions*.

➔ 7. Agora, a página inicial do setor de suporte lista o assunto como *Answered* (respondido).

Para mais detalhes sobre como criar um aplicativo baseado nos formulários do TWiki, consulte a documentação correspondente em [12].

Cada tópico pode conter um ou mais arquivos de qualquer tipo, bastando usar a tela *Attach* [13] (item 9 da [figura 1](#)).

Arquivos anexados ao conteúdo podem render uma solução personalizada de groupware, como por exemplo compartilhamento de arquivos ou sistemas de gerenciamento de documentação. Em nosso exemplo do suporte, um arquivo contendo uma correção para uma falha qualquer poderia ser a resposta para a reclamação do usuário. Ou seja, podemos organizar o conteúdo de forma lógica, com arquivos anexados aos assuntos. Não é preciso se preocupar com links quebrados ou um servidor FTP fora do ar: a própria estrutura da informação oferece os arquivos corretos na hora certa.

Todos os assuntos e arquivos anexados são catalogados em um sistema de controle de versões (item 10 da [figura 1](#)). Com esse controle, sempre se pode rastrear quem fez o quê, quem mudou o quê e quando, e as versões mais antigas dos documentos podem ser recuperadas a qualquer tempo. O controle de versões é absolutamente invisível ao usuário, já que ocorre automaticamente – não há nenhuma tela para isso. Uma coisa bacana do TWiki é que é possível fazer edições repetidas em assuntos sem que cada vez que você o salvar uma nova

versão seja criada. Se as alterações estiverem dentro do período de uma hora a partir do início, todas são salvas na mesma versão. Bastante útil para “ver

como está ficando” e corrigir erros de digitação sem gerar montes de versões do mesmo documento.

O TWiki também oferece controle de acesso [14]. Podemos restringir o acesso a cada um dos assuntos e até mesmo a webs inteiras. Cada usuário tem seu próprio conjunto de permissões e os grupos de usuários também têm os seus. As permissões são divididas em três áreas: visitação (consulta), edição e troca de nome. O controle de acesso permite a construção de um sistema com “castas” bastante definidas e flexíveis.

O espírito wiki reside na liberdade de inserir qualquer tipo de dado, sem restrições quanto a formato (ou formulários castradores) e censuras de qualquer tipo. Por essa razão, recomendamos fortemente que você pense bem antes de sair restringindo o acesso de leitura ou mesmo de escrita em seu TWiki. Quanto mais restrições forem impostas, menos vantagens podemos retirar de um wiki.

Plugins para o TWiki

O sistema de plugins do TWiki permitem estender a funcionalidade do programa. Os usuários não precisam instalar absolutamente nada – o único com poderes para isso é o administrador, que o faz no servidor em que o TWiki está instalado. Um repositório bem sortido e cujo acervo cresce a cada dia [17] está hospedado no site oficial twiki.org – mais de

Listagem 1: Exemplo com o ChartPlugin

```
01 %TABLE{name="exampleTable2"}%
02 Year 1998 1999 2000 2001 2002
03 Low 8 10 13 17 22
04 Target 10 12 17 20 25
05 High 12 15 19 24 29
06 Actual 9 14 15 19 27
07
08 %CHART{ type="combo" subtype="bar, pline, ?
point, line" name="combo2" table="exampleTable2" ?
data="R2:C2..R5:C6" xaxis="R1:C2..R1:C6" ?
legend="R2:C1..R5:C1" width="225" height="200" }%
```

Listagem 2: Quadro de atendentes, 2a. iteração

	Início	Fim	Responsável	Substituto
01	07:00	11:00	RicardoFraga	
02	11:00	15:00	PedroTonelli	
03	15:00	19:00	SandraCarvalho	

140 plugins estão disponíveis. Os plugins são uma maneira bem jóia de deixar o TWiki tinindo, ajustado perfeitamente a suas necessidades – como um terno feito por um alfaiate. Alguns dos plugins mais populares incluem:

- ➔ *ActionTrackerPlugin*: permite rastrear ações dentro de intervalos de tempo pré-determinados e avisar aos interessados por email.
- ➔ *CalendarPlugin*: usado para criar um calendário mensal, com agenda de compromissos integrada;
- ➔ *ChartPlugin*: um útil criador de diagramas, baseado nas tabelas do TWiki;
- ➔ *DatabasePlugin*: permite acesso direto aos bancos de dados em uso;
- ➔ *EditTablePlugin*: um utilíssimo editor de tabelas TWiki, com todos os tipos de campos, opções e controles que você poderia imaginar;
- ➔ *HeadlinesPlugin*: mostra notícias fresquinhas, que recebe de sites na Internet pelo protocolo RSS;
- ➔ *TablePlugin*: controla atributos de tabelas e implementa a classificação (*sorting*) por colunas;
- ➔ *TWikiDrawPlugin*: um applet em Java para criar desenhos dentro de assuntos. Funciona como um quadro-branco;
- ➔ *SlideShowPlugin*: transforma a página TWiki selecionada em uma apresentação em HTML à la PowerPoint.

O *SpreadsheetPlugin* é instalado por padrão no TWiki. Com ele, é possível criar planilhas no wiki. Mais de 60 fórmulas estão disponíveis, como por exemplo

`$AVERAGE()`, `$IF()`, `$REPLACE()`, `$TIME()`, `$SET()` e `$GET()`. Uma fórmula pode ser colocada em qualquer lugar dentro de um assunto – ou seja, não precisa estar necessariamente inserida em uma tabela. Por exemplo, `%CALC{"$PROPERSPACE(LinuxMagazine?)"}%` retorna o valor *Linux Magazine*. Sistemas baseados em web podem usar a fórmula `$IF()` para operações condicionais – e, com elas, dá para fazer muita coisa.

O *ChartPlugin* cria diagramas em PNG ou GIF para visualizar tabelas TWiki em escala, que pode ser linear ou semi-logarítmica. A **listagem 1** mostra um exemplo de uso desse interessante plugin, cujos resultados mostramos na **figura 4**.

Sistemas simples para a web

Feche os olhos, relaxe e respire bem fundo. Pronto? Agora imagine a seguinte situação: sua empresa possui um pequeno *call center* que presta suporte telefônico a clientes externos. O gerente que cuida de tudo isso mantém em uma planilha do *OpenOffice.org Calc* um registro dos tempos que os técnicos levam para atender aos usuários. Uma cópia impressa desta planilha é afixada na saleta do café para que todos possam ver; outra cópia é enviada ao chefe. Para simplificar o trabalho, o gerente encomenda um sistema eletrônico online que substitua essa planilha e suas cópias impressas. Todos devem ser capazes de ver quem estava atendendo chamados e em que período e o “estado de operação” (*ocupado, livre,*

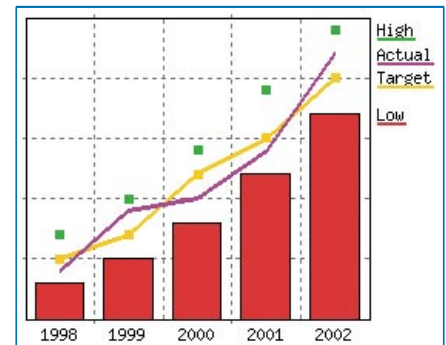


Figure 4: Diagrama resultante da fórmula da listagem 1.

tomando café) de cada atendente deve poder ser facilmente alterado e instantaneamente mostrado.

Um sistema baseado num wiki é normalmente feito por iterações – comece pequeno e vá aumentando a complexidade aos poucos conforme necessário. O bacana é que enquanto você desenvolve uma iteração, a anterior já está no ar. A primeira iteração é a criação de um tópico com marcadores simples:

```
* 07:00 - 11:00: Ricardo
* 11:00 - 15:00: Pedro
* 15:00 - 19:00: Sandra
```

Um “assunto”, como são chamadas as páginas do TWiki, preenche todos os requisitos para ser a interface com os usuários do sistema proposto: qualquer um pode ver quem está “de serviço” no momento e pode, inclusive, atualizar o quadro. Vamos, agora, tornar a apresentação dos dados mais útil e agradável. Para isso, que tal converter os marcadores em

Listagem 3: Quadro de atendentes, 3a. iteração

```
01 %EDITTABLE{ header="on" format="| select, 1, , 07:00, 07:30, 08:00, 08:30, 09:00, 09:30, 10:00, 10:30, 11:00, 11:30, 12:00, 12:30, 13:00, 13:30, 14:00, 14:30, 15:00, 15:30, 16:00, 16:30, 17:00, 17:30, 18:00, 18:30, 19:00 | select, 1, , 07:00, 07:30, 08:00, 08:30, 09:00, 09:30, 10:00, 10:30, 11:00, 11:30, 12:00, 12:30, 13:00, 13:30, 14:00, 14:30, 15:00, 15:30, 16:00, 16:30, 17:00, 17:30, 18:00, 18:30, 19:00 | select, 1, , ArturClemente, NicolasNabuco, PedroTorelli, RicardoFraga, SandraCarvalho, EstelaReiner | text, 20 | text, 40 |" changerows="off" }%
02 Início Fim Responsável Substituto Comentários
03 07:00 11:00 RicardoFraga
04 11:00 15:00 PedroTorelli
05 15:00 19:00 SandraCarvalho
```

uma tabela e usar WikiWords no nome dos funcionários? Assim, clicar no nome deles leva às suas páginas pessoais, que podem conter informações relevantes para a equipe. Vamos, além disso, adicionar uma coluna para um substituto caso o encarregado daquele horário tenha que levar a filha ao médico ([listagem 2](#)).

Histórias de sucesso com o TWiki

O site oficial do TWiki [\[3\]](#) permite que empresas que usem o sistema enviem “histórias de sucesso” descrevendo suas experiências. Entre as empresas que enviaram seus relatos incluem-se:

- ➔ Appropriate Solutions
- ➔ Astrogrid
- ➔ British Telecom
- ➔ Cingular Wireless
- ➔ Cmed
- ➔ Disney
- ➔ Ensequence
- ➔ Helsinki Institute of Physics
- ➔ Honeybee Robotics
- ➔ Michelin
- ➔ Motorola
- ➔ Orbis Technology
- ➔ Portland State Aerospace Society
- ➔ Seal Systems
- ➔ Tempest Security Technology
- ➔ Wind River
- ➔ Yahoo

- 07:00 - 11:00 Uhr: Richard
- 11:00 - 15:00 Uhr: Peter
- 15:00 - 19:00 Uhr: Sam

Figura 5: A iteração 3 do quadro de atendentes.

Start	End	Primary	Backup	Comments
07:00am	11:00am	Main.RichardDonkin		
11:00am	03:00pm	Main.PeterThoeny		
03:00pm	07:00pm	Main.SamHasler		

Figura 6: A mesma iteração, vista em modo de edição.

Agora vamos tornar as coisas mais amigáveis para os usuários. Usaremos o *EditTablePlugin* para que os usuários possam selecionar os horários e nomes de uma lista (veja a [listagem 3](#)). O modo de visualização e de edição resultantes da [listagem 3](#) podem ser vistos na [figura 5](#) e [figura 6](#).

O quadro de atendentes do call-center é um sisteminha bastante simples que faz exatamente o que o cliente precisa. Aplicações mais sofisticadas podem ser criadas com os formulários TWiki, mecanismos de busca embutidos e algum HTML pra “grudar” tudo isso junto. Um exemplo é o diretório de instalação do próprio TWiki [\[18\]](#), que possui um formulário para envio de novas instalações, todas listadas em uma grande tabela. Um filtro pode ser aplicado à lista para, por exemplo, mostrar apenas as instalações empresariais que funcionam atrás de firewalls. Você é livre para “pegar emprestado” o código fonte desse formulário para criar seus próprios sistemas. Dica: para ver o código fonte de um assunto no TWiki, basta inserir `?raw=on` no final da URL.

Conclusão

O TWiki é uma ferramenta poderosa para criar ambientes de colaboração via web. O conteúdo é livre e pode ser formatado e estruturado (ou não) a gosto do freguês. Se alguma estruturação for necessária, ela pode ser facilmente adicionada mais tarde. Nos primeiros meses de uso do sistema será necessário muito acompanhamento, palestras, conversa e material informativo para que a idéia seja bem aceita entre os usuários. Depois que ela “pegar”, entretanto, você verá que o crescimento do número de usuários será endêmico. O TWiki é a plataforma para construir sistemas web flexíveis de forma fácil e, acima de tudo, eficiente. ■

SOBRE O AUTOR

Peter é um desenvolvedor com 15 anos de experiência. Especialista em arquitetura de software, interfaces com o usuário e tecnologias da web, Peter é o autor do TWiki e mantenedor do projeto há mais de cinco anos. Ele também é o administrador do grupo de Engenharia do Conhecimento na Universidade de Wind River que, por acaso, possui uma das maiores bases instaladas de TWiki do planeta.

INFORMAÇÕES

- [1] Wikipedia: en.wikipedia.org/wiki/Main_Page
- [2] WikiWikiWeb: c2.com/cgi/wiki?WelcomeVisitors
- [3] Página oficial do TWiki: twiki.org/
- [4] Histórias de sucesso do TWiki: twiki.org/cgi-bin/view/Main/TWikiSuccessStories
- [5] Baixe o TWiki! twiki.org/download.html
- [6] Documentação sobre transmissões RSS: twiki.org/cgi-bin/view/TWiki/WebRssBase
- [7] Variáveis do TWiki: twiki.org/cgi-bin/view/TWiki/TWikiVariables
- [8] Variável %SEARCH{}%: twiki.org/cgi-bin/view/TWiki/TWikiSearch
- [9] Kupu, um editor WYSIWYG para o TWiki: twiki.org/cgi-bin/view/Plugins/KupuEditorAddOn
- [10] Temas para o TWiki: twiki.org/cgi-bin/view/TWiki/TWikiTemplates
- [11] Web de suporte do TWiki: twiki.org/cgi-bin/view/Support/WebHome
- [12] Formulários TWiki: twiki.org/cgi-bin/view/TWiki/TWikiForms
- [13] Anexos em assuntos TWiki: twiki.org/cgi-bin/view/TWiki/FileAttachment
- [14] Controle de acesso: twiki.org/cgi-bin/view/TWiki/TWikiAccessControl
- [15] Bem vindo, amigo! twiki.org/cgi-bin/view/TWiki/WelcomeGuest
- [16] Apresentação “um TWiki como tira-gosto”: twiki.org/cgi-bin/view/TWiki/ATasteOfTWiki
- [17] Repositório de plugins do TWiki: twiki.org/cgi-bin/view/Plugins/WebHome
- [18] Diretório de instalações do TWiki:

Estruturando comunidades

Conteúdo a conta-gotas

Os “frameworks” de desenvolvimento em PHP para sites dinâmicos tornam a vida de seus mantenedores muito mais simples. Um deles, que atende pelo nome de Drupal, é um forte candidato a ser o melhor de todos. Seu sistema modular é extensível e bastante poderoso e a aparência do site pode ser alterada em segundos usando modelos. Neste artigo, vamos mergulhar fundo na vasta coleção de funções do Drupal e discutir sua instalação e configuração básica.

POR UWE HERMANN

Os sites destinados a comunidades virtuais estão entre os mais visitados de toda a Internet. Os usuários podem registrar-se no site e enviar artigos, matérias e comentários a respeito de uma infinidade de assuntos. O Drupal [1] foi desenvolvido exatamente para essa tarefa e oferece tudo de que se possa precisar para colocar um site comunitário no ar.

Mas o Drupal extrapola sua definição. Pode-se dizer que é um CMS (*Content Management System* ou sistema de administração de conteúdo) capaz de gerenciar uma grande variedade de sites, desde blogs simples até sites profissionais para companhias. O Drupal foi escrito em PHP e usa um banco de dados para armazenar seu conteúdo. O código fonte do sistema está licenciado sob os termos da *GNU General Public License* (GPL) e pode portanto ser usado livremente, modificado sem limites e distribuído a seu bel-prazer. O quadro **O nascimento de Drupal** apresenta sua história e explica como ele ganhou esse estranho nome.

Descompacte e... pronto!

O Drupal não é difícil de instalar. Entretanto, não é assim tão direto como se poderia esperar, especialmente se você já conhece outros sistemas semelhantes de CMS. Como sempre (isso já está ficando chato...) os felizardos usuários do Debian (e derivados como Xandros, UserLinux e Ubuntu) podem, simplesmente, digitar `apt-get update && apt-get install drupal` e responder às telas de instalação. Se você é fã da interface gráfica, nada como o Synaptic para facilitar a sua vida. Agora, se você não gosta do Debian, esse luxo não está à sua disposição. Nesse caso, uma dolorosa caminhada pelas trilhas da instalação manual deve ser iniciada. Nada de arrepiar os cabelos, mas o processo requer alguns cuidados.

Os requisitos técnicos não têm nada que não seja feijão-com-arroz: um servidor web (interprete “servidor web” como sendo o *Apache*, mas até mesmo o *Microsoft IIS* irá funcionar em caso de emergência...) e o interpretador PHP na versão 4.1 ou superior. Na época em que este artigo foi para as rotativas, a versão 5 do PHP não era completamente compatível com o Drupal, embora os desenvolvedores estivessem trabalhando

O nascimento de Drupal

O belga Dries Buytaert [2], cientista da computação e autor original do Drupal, iniciou no ano 2000 o desenvolvimento de um sistema de CMS (*Content Management System* ou sistema de administração de conteúdo) para gerenciar sites com conteúdo dinâmico. O sistema deveria servir de base para um blog cooperativo de uma comunidade acadêmica, que Dries queria chamar de *Dorp* (palavra holandesa que significa “povoado”). Ao verificar a disponibilidade do domínio dorp.org, Dries tropeçou na digitação e acabou indo parar em drop.org. Gostou tanto do nome que decidiu registrar o domínio em seu nome e rebatizar o software – *Drupal* é a pronúncia de “gota” (*drop*) em holandês (escreve-se *Druppel*). Hoje, 5 anos depois, o Drupal cresceu e floresceu, tornando-se um CMS maduro e flexível, com centenas de desenvolvedores trabalhando para aprimorar e estender suas funções cada vez mais.

arduamente nesse sentido. É necessário ativar a opção `session.save_handler user` no PHP para que o Drupal funcione; os desenvolvedores recomendam, ainda, anular a limitação de cache com `session.cache_limiter none`. Ambas as opções são linhas no arquivo de configuração do PHP, o `php.ini`. Não tema: o Drupal oferece um arquivo `.htaccess` com essas opções pré-ajustadas (e de brinde, mais alguns *defaults* razoáveis).

Solução de problemas

Muitos dos problemas mais comuns de instalação são simples de evitar.

- Um dos erros mais comuns é atribuir o valor `localhost` para a variável `$base_url` no arquivo `includes/conf.php`. Esse valor é o correto se estivermos fazendo testes locais (como no exemplo deste artigo) mas **não** quando o Drupal estiver rodando em um domínio existente. Nesse caso, a linha conterá algo parecido com `$base_url = "http://www.meusite.com.br";`. Se seus DNSs ou domínios não estiverem funcionando corretamente, pode-se especificar um endereço IP nesse campo.
- Outro erro freqüente é especificar um caminho incorreto para o servidor. Se o Drupal não estiver instalado no diretório `/var/www`, mas em `/var/www/drupal`, por exemplo, a configuração correta seria `$base_url = "http://www.meusite.com.br/drupal";`.
- Não deve haver um caractere barra (/) no fim da variável `$base_url`. Em outras palavras, o correto é `http://www.meusite.com.br` e não `http://www.meusite.com.br/`.
- Se você hospeda seu Drupal em um provedor e ele já configurou um banco de dados no MySQL com usuário e senha, pule a etapa **GRANT** e as de criação do banco. Use os valores fornecidos pelo provedor em vez de `drupaluser`, `secret` e `drupal`.

Caso a instalação não funcione mesmo assim, pesquise a documentação oficial no site do Drupal [1]. Uma boa idéia é também “fuçar” nos fóruns [3] e listas de discussão [4], que são planícies onde reina um grupo de desenvolvedores sérios e competentes que ficarão muito felizes em ajudá-lo.

O Drupal precisa também de um banco de dados. Os gerenciadores reconhecidos são o MySQL versão 3 (preferencialmente numa versão mais recente que a 3.23.17) e versão 4, além do PostgreSQL. Para recursos que precisem de XML, como o RSS e a API do *Blogger.com*, o PHP vai precisar ter à mão suas extensões de XML, que na maioria das distribuições já estão instaladas por padrão.

O recurso *Clean URLs* (que mostra ao internauta uma URL “bonitinha”) precisa do módulo `mod_rewrite` no Apache e permissão de usar arquivos `.htaccess`. Para que isso funcione, evite configurar como `None` a diretiva `AllowOverride` no Apache para o diretório onde o Drupal está instalado. Com o *Clean URLs*, o Drupal pega um endereço feio como www.meusite.com.br?q=node/34, e o mostra ao internauta como www.meusite.com.br/node/34. Além de melhorar a navegação e permitir que o visitante do site guarde um endereço que faz mais sentido para ele, os mecanismos de busca (como o Google) poderão cadastrar suas páginas mais facilmente. Falando neles, é possível otimizar mais ainda a penetração do seu site nos Googles da vida, bastando ativar o módulo `path` (caminho), que permite associar URLs ainda mais “humanas” a cada um dos itens do seu site. Por exemplo, em vez de www.meusite.com.br/node/123, você poderia ter www.meusite.com.br/lojavirtual, muito mais fácil de lembrar.

Mesmo webmasters inexperientes com um conhecimento módico de Linux não terão problema algum para instalar o Drupal usando um cliente FTP e o utilitário *PhpMyAdmin*. As etapas que mostramos a seguir instalam o Drupal em uma máquina local ou em um provedor que dá acesso via SSH (*Secure Shell*) à sua conta. Depois de baixar de drupal.org a versão mais atual do programa (no momento em que editávamos este artigo era a 4.6.0), rode esses comandos para descomprimir o pacote e mover seu conteúdo para o diretório de documentos

(também chamado de *document root*) do servidor web – em nosso exemplo, este diretório é o `/var/www`:

```
tar xfvz drupal-4.5.2.tar.gz
mv drupal-4.5.2/* drupal-4.5.2/.htaccess 2
/var/www
```

Os comandos de MySQL a seguir, `mysqladmin` e `mysql`, criam e preparam um banco de dados para uso do Drupal, e criam um usuário dentro do banco:

```
$ mysqladmin -u root -p create drupal
Enter password:
$ mysql -u root -p
mysql> GRANT ALL PRIVILEGES ON drupal.* 2
TO drupaluser@localhost IDENTIFIED BY 'secret';
mysql> FLUSH PRIVILEGES;
```

Isso cria um banco de dados chamado `drupal`. O usuário do banco, chamado de `drupaluser`, cuja senha é `secret`, possui privilégios ilimitados em todas as tabelas desse banco. Pressione **[Ctrl]+[D]** para sair do shell do MySQL.

O arquivo `database/database.pgsql` no diretório de instalação do Drupal contém uma definição de esquema XML para uso com o PostgreSQL. O MySQL também tem um, que fica em `database/database.mysql`. Basta redirecionar o arquivo no shell para que seja interpretado:

```
$ mysql -u drupaluser -p drupal < /var/2
www/database/database.mysql
Enter password:
```

Se o `mysql` não reclamar nesse ponto da história, podemos considerar que até aqui tivemos sucesso. Para completar a instalação, precisamos inserir no arquivo `includes/conf.php` um item para controle de acesso do MySQL e o diretório onde o Drupal está instalado. Ponto para os desenvolvedores: esse arquivo é im-

pressionantemente fácil de ler e entender. As variáveis que precisamos alterar são `$db_url` e `$base_url`; os valores que atribuiremos a elas são, em nosso exemplo:

```
$db_url = "mysql://drupaluser:2secret@localhost/drupal";
$base_url = "http://localhost";
```

A primeira linha informa ao Drupal qual banco de dados deve usar, o nome do usuário, a senha, o nome da máquina que hospeda o banco de dados e o nome do próprio banco. A segunda linha especifica o endereço do Drupal para os internautas. Para nossos testes, por enquanto será <http://localhost>.

Aproveite e teste com seu navegador para ver se o Drupal está mesmo em <http://localhost>. A primeira coisa a fazer, agora, é configurar um usuário para o Drupal. Esse usuário será o responsável pela administração do sistema e terá poderes supremos e divinos sobre ele, tanto para administração de conteúdo como para configuração do programa. Qualquer nome serve, mas o tradicional é *admin*. Caso encontre alguma dificuldade ou se depare com algum erro durante o processo, não hesite em consultar o quadro [solução de problemas](#).

Tabela 1: Módulos intrínsecos ao Drupal	
Módulo	Descrição
aggregator	Permite captar a transmissão de notícias vindas de outros sites e mostrá-las em um bloco. O protocolo usado é o RSS.
ping	Caso o conteúdo mude, esse módulo notifica o fato a serviços como technorati.com , weblogs.com e blo.gs . Muito útil para blogs.
search	Mecanismo interno de busca. Precisa do <i>cron</i> para atualizar automaticamente o índice de pesquisa.
statistics	Gera estatísticas de visitação do site. Guarda o endereço IP do site de onde o visitante veio, o número de visitas e coisas desse tipo.
throttle	Esse módulo desabilita alguns dos outros módulos e blocos quando o tráfego é intenso, aumentando o desempenho e evitando que seu site seja "slashdotted". Depende do módulo <i>statistics</i> para funcionar.
watchdog	Registra todos os eventos do site: quem iniciou uma sessão, quem (e quando) criou ou modificou conteúdo, mensagens de erro etc.

Conceitos básicos sobre as goteiras

A terminologia do Drupal difere um pouco da dos outros sistemas de administração de conteúdo. As seções de agora em diante exploram as partes mais importantes do sistema Drupal e explicam os termos relevantes.

Um nó (*node*) é o "tijolinho" básico que será usado para construir um site no Drupal. Um nó pode ser um artigo, uma imagem ou um comentário num fórum. Os seguintes tipos de nó são habilitados por padrão: *page*, para páginas estáticas, e *story*, para artigos e matérias. É possível incluir mais tipos,

bastando para isso instalar e habilitar o módulo correspondente. Por exemplo, o módulo *forum* define um novo tipo de nó chamado *forum topic* (assunto em discussão); o módulo *blog* define um *personal blog entry* (anotação em blog pessoal). O menu de administração mostra uma lista de tipos de nós no item *create content* (criar conteúdo).

Um bloco é uma caixa com conteúdo arbitrário que aparece à esquerda ou à direita dos sites. O Drupal oferece um número razoável de blocos pré-configurados, como *User Login* (iniciar sessão de usuário), *Who's online* (quem está conectado agora) e *Recent Comments*

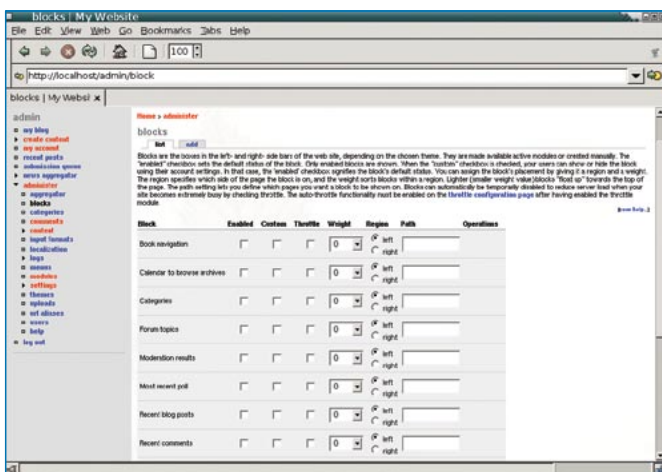


Figura 1: Os blocos oferecem informação adicional aos visitantes do site. Eles são mostrados à direita e à esquerda de todas as páginas (ou apenas nas páginas cujo endereço é especificado por expressões regulares [5]).

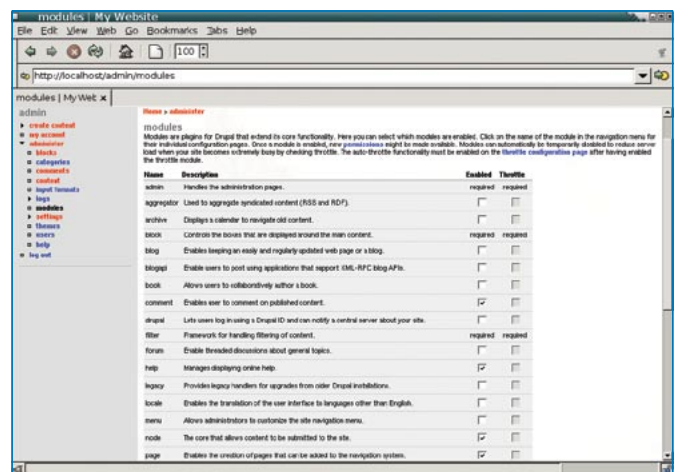


Figura 2: O Drupal é instantaneamente extensível com o uso dos incontáveis módulos disponíveis. São tantas opções que fica difícil encontrar algo que não possa ser feito com um ou dois módulos extras.

(comentários mais recentes). Além disso, qualquer novo módulo instalado pode oferecer mais opções de blocos, bastando o administrador habilitá-los (ver [figura 1](#)). O administrador pode ainda definir novos blocos que conttenham tanto código HTML como texto puro e, para conteúdo mais complexo, código em PHP e consultas (*queries*) SQL. O campo *Custom* em cada bloco especifica se o usuário que iniciou uma sessão no site pode ligar ou desligar blocos a seu bel-prazer, personalizando a aparência do site. No site oficial do Drupal há uma área chamada *Custom Blocks Repository* (repositório de blocos personalizados), onde podemos encontrar uma vasta coleção de blocos. Um deles certamente é justamente o que você está procurando.

O Drupal é, essencialmente, uma “ar-mação” onde os módulos são “parafusados”. Essa estrutura básica é bastante espartana, oferecendo apenas as funcionalidades mais críticas e primordiais. Isso dá ao sistema uma flexibilidade sem par (ver [figura 2](#)). Um módulo, tipicamente, é um pacote de rotinas em PHP contido num arquivo cujo nome é sempre *nome_do_modulo.module*. Pode haver alguns arquivos adicionais, como uma

folha de estilo CSS e imagens usadas pelo módulo. Na área de downloads do Drupal [6] encontramos um número gigantesco de módulos, cada um apropriado para uma função específica (ver [tabelas 1 e 2](#)).

A terminologia usada no Drupal precisa de alguma explicação, já que as coisas não têm os mesmos nomes padronizados que aparecem em outros gerenciadores de conteúdo. Usa-se as palavras *vocabulary* (vocabulário) e *term* (termo) para criar o que muitas pessoas por aí chamariam de *categoria*. Cada *vocabulário* tem um nome (como *Tópico*, por exemplo) e múltiplos termos relacionados a ele, como *Política*, *Religião*, *Esportes* e *Tecnologia*. Podemos definir o número de vocabulários para um site e associar nós a categorias (ou melhor, a *termos*) específicos, ligados a esses vocabulários. Esse esquema é bem flexível e aplicável a muitos contextos. Por exemplo, as hierarquias dentro de fóruns ou as galerias de fotos podem ser modeladas usando-se o sistema taxonômico do Drupal.

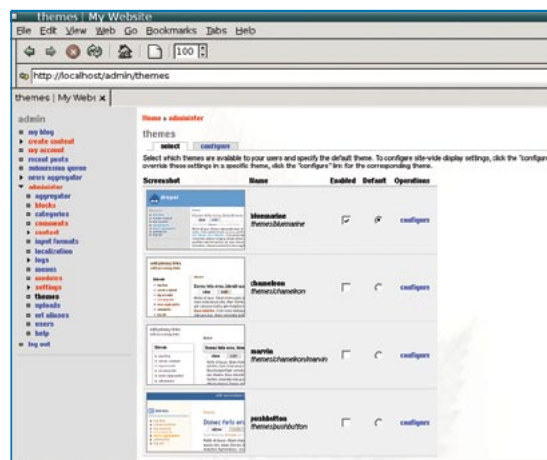


Figura 3: Os temas permitem que o administrador (e até os usuários normais, se desejado) modifiquem a aparência do site.

Os chamados *temas* permitem que o administrador apresente o usuário com “rostos” personalizados para o site. (ver [figura 3](#)). Um número assustador de temas prontos pode ser encontrado em [6], e não é difícil modificar um tema para seu próprio uso. O *Drupal Theme Garden* [7] lista alguns temas e deixa você modificá-los e testar seu funcionamento lá mesmo. É uma boa “caixa de areia”, não?

O Drupal implementa um sistema de administração dos direitos e permissões dos usuários, baseado na função ou cargo designado para cada um. Um usuário registrado (*user*) pode iniciar uma sessão no site e ter à sua disposição um certo número de controles, dependendo do seu cargo (*role*) e das permissões (*permissions*) dadas pelo administrador a esses cargos. Mais adiante discutiremos esse arranjo com mais atenção.

Criando conteúdo

Usuários registrados podem criar conteúdo (*create content*) e mesmo novos nós para abrigar esse conteúdo, tudo a partir das opções do menu. Dependendo dos módulos ativos, o conteúdo pode ser um artigo, mensagens em um blog, imagens, pesquisas de opinião e muitas outras coisas. A [figura 4](#) mostra como criar um nó chamado *page* (página). Cada nó deve

Tabela 2: Módulos opcionais do Drupal

Módulo	Descrição
<i>atom</i>	Gera uma transmissão de notícias no formato <i>atom</i> (versão 0.3); o conteúdo fica disponível em http://localhost/atom/feed .
<i>dba</i>	A minúscula ferramenta de administração de bancos de dados do Drupal. Pode poupá-lo do trabalho de usar ferramentas como o <i>PhpMyAdmin</i> e semelhantes.
<i>flexinode</i>	Permite a definição de quaisquer formatos de conteúdo. Por exemplo, é possível definir um pseudonó chamado <i>receita</i> , contendo um nome, ingredientes, quantidades, dicas de preparação e afins, tudo isso usando apenas o mouse.
<i>print</i>	Cria um link do tipo <i>Imprima esta página</i> para cada nó; uma página “amigável à impressora” é criada instantaneamente e pode ser impressa com economia de papel e tinta.
<i>trackback</i>	Brinquedo obrigatório para blogueiros. Permite que outros blogueiros enviem os chamados <i>trackbacks</i> (numa tradução livre, “rastreamento inverso”) para o site e que, em contrapartida, o site envie <i>trackbacks</i> a eles.
<i>scheduler</i>	Permite agendar a publicação de conteúdo. Basta definir a data e a hora que deve aparecer e o sistema se encarrega do resto.
<i>htmlarea</i>	O conhecidíssimo editor de HTML embutido, também usado em outros CMSs.

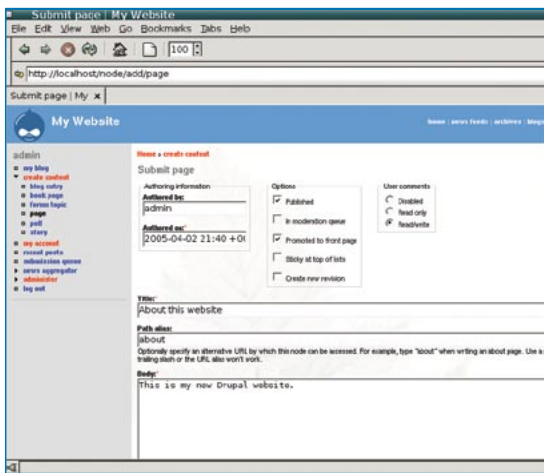


Figura 4: Não é por falta de opções que deixaremos de criar nós. O exemplo mostra um nó do tipo *page*.

ter, no mínimo, um título e um conteúdo, que chamamos de *body* (corpo). Se o módulo *path* estiver ativo, o usuário pode inclusive atribuir uma URL mais “decente” a esse nó.

Há muitas opções de configuração dos nós. Por exemplo, podemos permitir ou barrar comentários a respeito do conteúdo do nó, mostrar ou não o nó na página inicial ou se o nó deve sempre aparecer no topo da lista de nós, não importando sua idade (*sticky* ou “grudento”, uma propriedade herdada dos fóruns). A chave *published* permite que o nó seja publicado (ou retirado de circulação) a qualquer tempo. Os nós, mesmo que não mostrados no site, continuam no banco de dados e podem ser republicados sempre que desejado. Para excluir definitivamente um nó, clique em *delete* na página *edit*.

Configuração básica

Os ajustes mais críticos estão disponíveis no menu de administração em *administer | settings* (*administrar | configurações* – ver **figura 5**). Aqui configuramos o nome do site, um slogan opcional, o email do administrador e um rodapé para as páginas. O campo *Default front page* define a página que será mostrada primeiro quando o visitante entra em seu site. Por padrão, ela é um nó (*node*); em

outras palavras, uma lista de artigos ou nós mais recentes (mas apenas aqueles artigos e nós que foram “promovidos” à página inicial usando a opção *Promoted to front page*). Não há artigos armazenados logo depois que o Drupal é instalado; em vez disso, o sistema apresenta um texto introdutório para o administrador.

O mecanismo de cache do Drupal melhora o desempenho do conjunto. Todas as páginas acessadas por usuários anônimos (ou seja, usuários que não informaram nome e senha) são guardadas no banco de dados para evitar que sejam geradas dinamicamente; ou seja, normalmente as páginas não existem de verdade, mas são geradas no instante em que o usuário clica no link. Como os usuários anônimos acessam umas poucas páginas (porque não fizeram *login*) mas o fazem um número absurdo de vezes (porque há muito mais anônimos que registrados) o Drupal já deixa as páginas “públicas” geradas de antemão e guardadas no banco de dados. Esse menu também pode ser usado para ativar a função *Clean URLs*.

Alguns dos módulos introduzem suas próprias telas de configuração no menu *administer | settings*; por exemplo, o módulo de estatísticas insere um item *administer | settings | statistics* no menu. Não vamos entrar em detalhes sobre isso nesse artigo. Para mais informações consulte o manual oficial do Drupal [8].

Tarefas repetitivas

O Drupal pode usar o sistema *cron* para tarefas recorrentes ou, ainda, chamadas periódicas à URL <http://www.meusite.com.br/cron.php>.

Ela dispara ações recorrentes no Drupal, como por exemplo a exclusão de registros muito antigos ou a atualização dos índices do mecanismo de busca. A página pode ser acessada com qualquer navegador, mesmo os de texto puro como o *wget* ou o *lynx*. A seguinte linha, se incluída no */etc/crontab*, dá conta do recado:

```
0 * * * * root /usr/bin/wget -o /dev/null -O /dev/null http://localhost/cron.php
```

Este exemplo acessa *cron.php* uma vez por hora (observe que a URL deve ser o domínio do site e não *localhost* ou *127.0.0.1* – mesmo que o cron esteja rodando na mesma máquina local). Se você não tem acesso ao cron (coisa comum em sites hospedados em provedores brasileiros) não se preocupe; sempre se pode lançar mão do módulo *poormanscron* [9] (algo como “cron de pobre”). Veremos mais adiante como usar esse módulo.

Administração de usuários

Também em *administer | users* podemos obter uma lista com os usuários registrados; os administradores podem configurar as preferências de cada um clicando em *edit*. Por exemplo, se um usuário se comportar mal (“mau sapão!”) é possível desabilitar essa conta. Os administradores

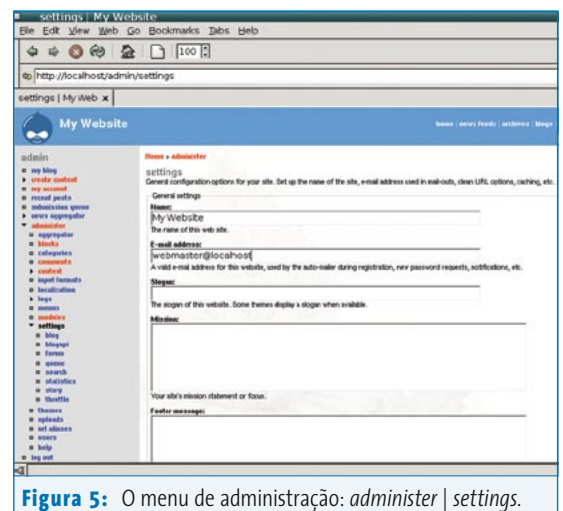


Figura 5: O menu de administração: *administer | settings*.

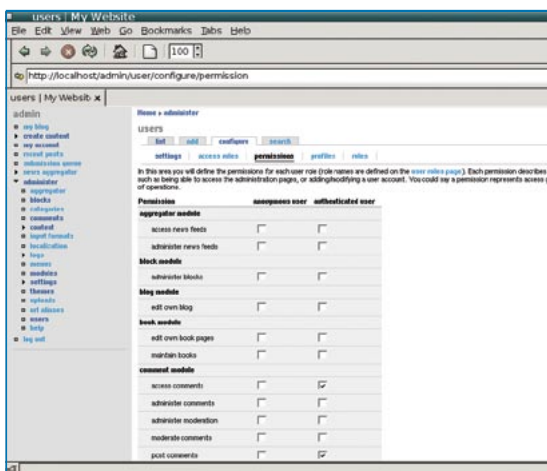


Figura 6: Cada cargo (aqui mostramos os usuários anônimos e os registrados) obedece a um conjunto de permissões. Cada módulo pode incluir um tipo adicional de permissões.

podem usar os submenus disponíveis em *administer* | *users* | *configure* para criar novos usuários e definir regras genéricas ou privilégios para as contas existentes.

➤ *settings*: Administradores podem usar esse item para especificar quem pode criar novos usuários. Em um site comunitário, os próprios usuários podem se inscrever. Em outras situações, cada conta individual tem que ser habilitada manualmente. O administrador pode também permitir o envio de imagens por parte dos usuários, que serão mostradas nas discussões dos fóruns.

➤ *access rules*: Os administradores podem proibir alguns nomes de usuário ou endereços de email (se contiverem, por exemplo, palavras obscenas). Para isso, podem definir uma ou mais regras contendo os coringas % (representando zero ou mais ocorrências) e _ (exatamente um caractere).

➤ *roles*: Os usuários podem ser nomeados para um ou mais cargos (*roles*). Os cargos padrão são *anonymous user* para usuários anônimos (ou seja, que não informaram nome e senha) e *authenticated user* para usuários que iniciaram uma sessão registrada. O administrador pode definir qualquer número de cargos adicionais e atribuir qualquer nome a

eles. Nomes bastante comuns para esses cargos são *administrador do fórum*, *editor*, *redator*, *revisor*, *cliente* e *parceiro*.

➤ *permissions*: A cada cargo está associado um conjunto de permissões, definidas pelo administrador. Por exemplo, *post comments* permite que os usuários postem comentários – e apenas isso. Teoricamente, qualquer módulo pode adicionar novas permissões à lista (ver **figura 6**). Por exemplo, o módulo *poll* (enquete) inclui uma permissão de criar novas pesquisas de opinião (*create*

polls) e outra para permitir que os usuários votem (*vote on polls*).

Essas opções tornam o sistema extremamente flexível, apto a resolver as mais diversas situações, problemas e cenários, desde um site de notícias como o Slashdot, no qual os usuários podem comentar livremente a respeito de notícias publicadas de forma controlada, até uma intranet empresarial com permissões diferentes para os gerentes, os diretores, os parceiros e para o “povão” do chão da fábrica.

Instalando novos módulos

O Drupal não é um sistema monolítico. Em vez disso, é uma armação bastante vazia que pode ser estendida até o espaço sideral com a adição de módulos. A instalação padrão traz um conjunto básico (mas ainda assim funcional) de módulos, embora nem todos estejam ativos. Mais de cem módulos opcionais podem ser baixados do site do Drupal, todos eles criados pela comunidade de desenvolvedores e cuidadosamente guardados num repositório CVS [10]. Obviamente, todos são livres e estão sob a licença GPL.

Instalar um módulo, entretanto, não é assim tão fácil quanto se poderia esperar. Pior: cada módulo pode ter um método diferente de instalação. Há um

procedimento padrão, que todos eles estão adotando, mas nem todos os módulos o seguem. Vamos instalar, como exemplo, os módulos *poormanscron* e *image*.

O *poormanscron* é um substituto para o *cron*. O Drupal inicia o módulo sempre que uma página é acessada para verificar se um dado intervalo de tempo (configurável) já terminou. Se sim, as ações definidas na página *cron.php* serão executadas e o contador de tempo volta a zero e reinicia a contagem. Esse módulo é bastante útil se seu provedor não dá acesso ao *cron* da máquina onde o site está hospedado. Os passos para sua instalação são simples:

```
$ wget http://drupal.org/files/projects/poormanscron-4.5.0.tar.gz
$ tar xfvz poormanscron-4.5.0.tar.gz
$ cd poormanscron/
$ cp poormanscron/LICENSE.txt poormanscron/
$ cp poormanscron/README.txt poormanscron/
$ cp poormanscron/poormanscron.module poormanscron/
```

O módulo é composto por um único arquivo, *poormanscron.module*, que deve ser simplesmente copiado para o diretório *modules* na sua instalação do Drupal. Depois, habilite o módulo no menu *administer* | *modules*. As configurações podem ser encontradas em *administer* | *settings* | *poormanscron*.

O módulo *image* é interessante para a maioria dos sites, pois implementa um gerenciador de imagens que permite, inclusive, que os usuários enviem fotos e gráficos para o site. O módulo requer as bibliotecas *ImageMagick*, *GD* e *ImLib2*, presentes na maioria das distribuições Linux. Pelo menos uma delas tem que estar disponível no provedor. Depois de baixar e descompactar o pacote *.tar*, o administrador precisa modificar o banco de dados no arquivo *image.sql*. Depois, o arquivo deve ser copiado para o diretório *modules*:


```
$ wget http://drupal.org/files/projects/2
image-4.5.0.tar.gz
$ tar xfvz image-4.5.0.tar.gz
[...]
$ mysql -u drupaluser -p drupal < image/2
image.sql
$ mkdir /var/www/modules/image
$ cp image/image.module image/image.inc 2
/var/www/modules/image
```

O próximo passo é ativar o módulo no menu *administer* | *modules* e entrar no menu *administer* | *settings* | *image* para configurá-lo. Por exemplo, o módulo precisa saber em quais diretórios deve gravar as imagens e miniaturas e, possivelmente, o caminho para o programa *convert*, parte do pacote ImageMagick. Normalmente ele está em */usr/bin/convert*, mas isso pode variar dependendo de sua instalação.

Também é possível criar galerias de imagem no menu *administer* | *categories*. Para isso, crie um vocabulário com um ou mais termos. Depois, defina o vocabulário como *Gallery Navigation Vocabulary* (vocabulário para navegação de imagens, novamente em *administer* | *settings* | *image*). Finalmente, selecione no menu *administer* | *users* | *configure* | *permissions* os usuários e cargos que têm permissão para enviar (e manipular) imagens. Com isso os usuários podem usar imagens nos conteúdos que criam (menu *create content*).

Expectativa

A versão mais recente do Drupal, a 4.6, foi lançada dia 15 de abril deste ano e traz um número bastante grande de mudanças interessantes. A nova versão possui um mecanismo de busca bastante aprimorado, que usa codificação UTF-8 e pode, portanto, trabalhar em praticamente qualquer idioma. A usabilidade da interface de busca, a abrangência da pesquisa e a relevância dos resultados estão muito mais avançadas. Os menus

de configuração também estão mais fáceis de usar, tornando mais simples o trabalho do administrador. O novo módulo *contact* (contato) permite que usuários registrados mandem mensagens de email a outros usuários registrados usando um formulário. É possível, agora, trabalhar com múltiplos sites sem ter que instalar um Drupal para cada um deles. Os gargalos de desempenho foram minimizados, portanto o Drupal consegue agora rodar com mais eficiência em situações de alto tráfego. O número, a qualidade e completude da interface de administração continuam a crescer: são mais de vinte idiomas já disponíveis. Graças ao comprometimento de alguns desenvolvedores, novos temas também estão disponíveis, incluindo os populares Kubrick, Persian e Manji, portados a partir do Wordpress [11]. Outras melhorias: *syndication* por RSS versão 2.0, o serviço *Ping-O-Matic*, que “pinga” todos os servidores de ping mais populares e o esperado suporte a PHP5. Para uma lista mais abrangente, consulte o *changelog* em [12].

O Drupal é um gerenciador de conteúdo gratuito, livre, compacto e altamente flexível. Graças a isso, uma megacomunidade se formou em torno dele, abrangendo milhares de usuários e desenvolvedores. O software pode ser facilmente adaptado para uma variedade bastante grande de aplicações. Os desenvolvedores apreciam sobremaneira a estrutura sólida e sua bem documentada interface de desenvolvimento (API - *Application Programming Interface*) [13], que é extensível usando módulos escritos em PHP. A “armação” modular, o mecanismo de cache e o módulo “afogador” (*throttle*) fazem do Drupal um produto altamente escalável – por exemplo, o site oficial, em drupal.org, possui nada menos do que dezoito mil usuários registrados e mais de dezessete mil nós criados. Sem dúvida um número impressionante, não acha? ■

SOBRE O AUTOR

Uwe Hermann está para se graduar em Ciência da Computação na Universidade Tecnológica de Munique. É fã ardoroso de Software Livre – tanto que, em seu tempo livre, em vez de jogar futebol, Uwe auxilia no desenvolvimento do Drupal e toma conta de projetos agonizantes [14] ou já defuntos [15]. Desnecessário dizer, portanto, que Uwe usa o Drupal no seu blog e em sua página pessoal [16].

INFORMAÇÕES

- [1] Site oficial do Drupal: www.drupal.org
- [2] Site oficial de Dries Buytaert, autor do Drupal: www.buytaert.net
- [3] Fóruns oficiais sobre o Drupal: drupal.org/forum
- [4] Listas de discussão sobre o Drupal: drupal.org/mailling-lists
- [5] Verbete na enciclopédia online Wikipedia sobre expressões regulares: www.wikipedia.org/wiki/Regular_Expression
- [6] Página de downloads do Drupal: drupal.org/project/releases
- [7] Drupal Theme Garden: webschuur.drupaldevs.org
- [8] Manual Oficial do Drupal: drupal.org/handbook
- [9] Módulo “cron de pobre”: drupal.org/project/poormanscron
- [10] Repositório de contribuições do Drupal (CVS): cvs.drupal.org/viewcvs/contributions/
- [11] Wordpress, software gratuito e livre para criação e manutenção de blogs: wordpress.org
- [12] Novidades no Drupal 4.6: drupaldocs.org/CHANGELOG.txt
- [13] Documentação da API do Drupal 4.5.x: drupaldocs.org/api/4.5
- [14] *Crazy Hacks*, site que é um verdadeiro asilo de projetos moribundos: www.crazy-hacks.org
- [15] Projetos livres já falecidos em busca de ressurreição (novos mantenedores): www.unmaintained-free-software.org
- [16] Página pessoal e blog de Uwe Hermann, mantida com o Drupal: www.hermann-uwe.de

Desenvolvendo aplicativos para a web com o Zope X3

Fórmula X



O programa de código aberto Zope, um servidor de aplicações web escrito em Python, é uma plataforma de gerenciamento de conteúdo extremamente popular. A versão X3.0, recentemente desenvolvida, acaba de ser lançada. O que será que há de novo?

POR PHILIPP VON WEITERSHAUSEN

Desde o lançamento do sistema de gerenciamento de conteúdo (ou CMS, *Content Management System*) Plone [1], é difícil imaginar um mundo de servidores livres de aplicações web sem o Zope [2]. O sucesso do projeto vem mostrar que o Zope pode facilmente ganhar terreno na batalha contra o Java, mesmo estando no campo desse último. A linguagem de programação Python traz um código fonte modular e orientado a objetos ao mundo do Zope, junto com a agilidade e a flexibilidade associadas mais tipicamente a linguagens de script web.

Do CMF a novos desenvolvimentos

O Zope é uma popular plataforma de CMS. A maioria dos sistemas CMS livres baseados no Zope (como Plone [1], Silva [3] e CPS [4]) usam um produto do Zope conhecido como Estrutura de Gerencia-

mento de Conteúdo (*Content Management Framework* – CMF). A estrutura CMF do Zope 2 não apenas tinha as ferramentas necessárias para desenvolver um sistema de gerenciamento de conteúdo, como também oferecia uma arquitetura baseada em componentes que permitia a modificação flexível de componentes individuais. A experiência ganha no desenvolvimento do CMF foi incorporada ao processo de desenvolvimento do Zope.

Em vez de retrabalhar o labiríntico código fonte do Zope, a Zope Corporation optou por começar do zero há três anos. Por causa de seu firme compromisso com o princípio do Software Livre, o Zope sempre teve o apoio de uma forte comunidade. Programadores experientes e novatos do Zope organizaram sessões de desenvolvimento grandes e pequenas (chamadas *sprints*, corridas de explosão) por todo o mundo para manter o projeto

em movimento. Métodos como *Xtreme Programming* (XP) e o teste de unidades (*Unit Testing*) trouxeram uma garantia de qualidade ao projeto. Além disso, o fato de o Zope versão 2 ser um produto estável e bem-sucedido deu aos desenvolvedores tempo bastante para otimizar recursos chave em múltiplos passos de iteração. E ninguém se queixou por ter sido preciso cavar tantos fossos no código para abrir caminho a melhores implementações.

Quadro 1: Entendendo o X

Quando o trabalho de desenvolvimento do Zope 3 foi iniciado, os projetistas rapidamente perceberam que teriam que abandonar qualquer esperança de ter uma API compatível com a do Zope 2. O prefixo “X” nas versões antigamente indicava que ela era “experimental”. Obviamente, a versão estável X3.0 do Zope é qualquer coisa, menos instável. O “X”, entretanto, continuou no nome para deixar claro que o X3 não é compatível com os anteriores.

Recursos

O núcleo do Zope X3 é a arquitetura de seus componentes. Diferentes componentes receberam a responsabilidade por tarefas específicas, como armazenamento de dados, processamento e apresentação.

Muitos leitores devem ter familiaridade com o ZODB (*Zope Object Database*, Banco de dados de objetos do Zope) por causa do Zope 2. O ZODB permite que os objetos permaneçam bastante transparentes no banco de dados e, ao mesmo tempo, oferece recursos empresariais como rastreamento de transações, auditoria e uma interface para o subsistema de armazenamento. O ZEO (*Zope Enterprise Objects*, objetos empresariais do Zope) permite mesmo o agrupamento de diversas instâncias do Zope, o que facilita a escalabilidade.

Um sistema de segurança flexível atribui permissões para proteger componentes, propriedades e métodos. Os usuários que queiram acessar componentes protegidos precisam de autorização. Componentes modulares autenticam e autorizam usuários, dando aos operadores a capacidade de adaptar a postura de segurança sem ter que refazer o aplicativo.

O Zope não só tem as ferramentas de localização necessárias aos desenvolvedores para permitir a internacionalização de aplicativos: o próprio Zope é plenamente internacional. Uma rica gama de recursos, como SMTP ou serviços de email baseados no *sendmail*, um sistema de notificação de eventos e o suporte a XML RPC vêm completar a paleta do Zope.

Mudança de paradigmas

O Zope X3 não tem alguns dos problemas que afetavam o Zope 2. Por exemplo, a versão anterior esperava que a instanciação de classes fornecesse os atributos e métodos necessários para interagir com o Zope. Isso fazia com que os desenvolvedores precisassem sobrecarregar os objetos com uma grande variedade de métodos,

o que limitava a portabilidade e dificultava muito mudanças subsequentes nas funções.

O Zope X3, por outro lado, mantém os componentes individuais tão simples quanto possível e adiciona componentes se for necessário adicionar funcionalidade. A arquitetura do Zope engloba vários tipos de componentes:

- ➔ Os componentes de *Content* (*Conteúdo*) tipicamente não têm métodos, possuem apenas propriedades que usam para publicar os dados armazenados. Tipos e valores de dados são especificados normalmente com um esquema de dados.
- ➔ *Utilities* (Utilitários) são componentes independentes de contexto que realizam uma tarefa específica, como conexões a bancos de dados, indexação ou entrega de email.
- ➔ *Adapters* (Adaptadores) são provavelmente os componentes mais poderosos. Eles permitem que os desenvolvedores adicionem funcionalidade a componentes existentes sem a necessidade de modificá-los. Essa técnica é extremamente útil, já que uma estrutura requer uma API específica. Os desenvolvedores podem deixar o componente original como está e implementar um adaptador que faça a interface entre o componente e a API necessária.
- ➔ As *Views* (Visualizações) mostram os outros componentes para o usuário. Um navegador web seria um bom exemplo, pois usa visualizações que renderizam páginas HTML. Uma visualização é na verdade um tipo especial de adaptador que fornece aos outros objetos um recurso que eles não possuem nativamente (apresentação).

Listagem 1: Interfaces (*interfaces.py*)

```
01 from zope.interface import Interface
02 from zope.schema import Text, TextLine
03
04 class IBuddy(Interface):
05     """Informacao sobre um colega"""
06
07     first = TextLine(title=u"nome")
08     last = TextLine(title=u"sobrenome")
09     address = Text(title=u"endereco")
10     zipcode = TextLine(title=u"CEP")
```

Contrato abstrato

Para permitir que os componentes permaneçam independentes da implementação, eles não são organizados por classe. Em vez disso, as interfaces são usadas para descrever a funcionalidade oferecida por um componente. Interfaces são um tipo de contrato formal que garante a provisão de uma função específica na forma de uma API. Como o Python não pode usar interfaces, o Zope as implementou ele mesmo.

A **listagem 1** mostra uma interface de um aplicativo de exemplo que gerencia endereços particulares. A interface mostrada aqui é um esquema de dados que descreve os componentes do conteúdo – neste caso, os dados do endereço de um amigo. A armazenagem do CEP permite buscar automaticamente as informações de cidade e estado mais tarde. ➔

Listagem 2: Componentes de conteúdo (*buddy.py*)

```
01 from persistent import Persistent
02 from zope.interface import implements
03 from buddydemo.interfaces import IBuddy
04
05 class Buddy(Persistent):
06     implements(IBuddy)
07
08     def __init__(self, first='', last='',
09                 address='', zip_code=''):
10         self.first = first
11         self.last = last
12         self.address = address
13         self.postal_code = postal_code
```


Listagem 3: Configuração (configure.zcml)

```

01 <configure
02     xmlns="http://namespaces.zope.org/zope"
03     xmlns:browser="http://namespaces.zope.org/browser">
04
05 <content class="buddydemo.buddy.Buddy">
06     <require
07         permission="zope.View"
08         interface="buddydemo.interfaces.IBuddy" />
09     <require
10         permission="zope.ManageContent"
11         set_schema="buddydemo.interfaces.IBuddy" />
12 </content>
13
14 <browser:addform
15     schema="buddydemo.interfaces.IBuddy"
16     label="Inserir o endereço de um camaradinho"
17     content_factory="buddydemo.buddy.Buddy"
18     arguments="first last address zipcode"
19     name="AddBuddy.html"
20     permission="zope.ManageContent" />
21
22 <browser:editform
23     schema="buddydemo.interfaces.IBuddy"
24     label="Editar o endereço do companheiro"
25     name="edit.html"
26     menu="zmi_views" title="Edit"
27     permission="zope.ManageContent" />
28
29 <browser:addMenuItem
30     class="buddydemo.buddy.Buddy"
31     title="Buddy"
32     permission="zope.ManageContent"
33     view="AddBuddy.html" />
34
35 </configure>

```

Na **listagem 1**, a declaração `class` do Python é usada para definir uma interface, já que o Python não tem interfaces nativamente. Além disso, o Zope não distingue entre uma interface que usa métodos para descrever funcionalidade e uma interface que define um esquema de dados.

Componentes simples de conteúdo

A tarefa de escrever uma classe persistente no Zope 2 era bastante complexa. No mínimo, eram precisos o metatipo, declarações de segurança e métodos de

instanciação para gerar as novas instâncias requeridas pela interface Web. Você ficará satisfeito em saber que agora as exigências são mínimas. Objetos persistentes só precisam lidar com os dados passados a eles; tudo o mais é manejado pelos outros componentes.

A **listagem 2** mostra uma implementação funcional das interfaces `IBuddy` mostradas na **listagem 1**. Note que a classe herda de `Persistent` para assegurar que suas instâncias sejam automaticamente armazenadas no ZODB. Para que essa classe se sobreponha sobre os

outros componentes, também é preciso especificar que a classe implementa a interface `IBuddy`.

Configuração em XML

O Zope 2 precisava importar bibliotecas do diretório `Products`. O módulo de inicialização para cada pacote (`__init__.py`) costumava conter o registro do componente. Outros elementos, como declarações de segurança ou configurações de visualização do navegador, precisavam ser definidos e implementados no código do aplicativo final.

O Zope X3 adota uma abordagem diferente. Por exemplo, as extensões do Zope agora são apenas pacotes simples em Python, o que significa que você pode instalá-los quando bem lhe aprouver – desde que estejam no `PYTHONPATH`. Tudo o mais que seja relacionado com configuração de componentes, como o próprio registro, declarações de segurança ou visualização do navegador, agora está contido no arquivo de configuração. Esse método dá aos desenvolvedores uma importante vantagem: a capacidade de desabilitar componentes temporariamente ou permanentemente sem modificar o código.

A **listagem 3** mostra as diretivas típicas de configuração para um componente de conteúdo baseado em *esquemas* (*schema*); o exemplo mostra apenas as declarações de segurança para os dados de leitura e escrita para as instâncias de objetos “buddy” – ou seja, cada um dos seus amigos. O arquivo então passa a definir dois formulários: um deles gera objetos “buddy”, o outro é usado para editá-los.

O Zope tem a capacidade de gerar automaticamente um formulário para o esquema de dados definido na interface `IBuddy`. Assim – como mostrado na **figura 1** – um esquema, uma implementação persistente simples, algumas diretivas de configuração e nenhum código em HTML dão forma a um componente que roda num navegador.

A última diretiva na [listagem 3](#) adiciona um item ao menu da interface web do Zope, item esse que permite aos usuários criar novos amiguinhos. O fato de essa diretiva existir exemplifica um importante princípio básico da filosofia do Zope 3: “Explícito é melhor que implícito”. Embora o desenvolvimento de software com o Zope 3 possa significar mais digitação, ao menos será mais fácil ler o código depois de passados seis meses – principalmente se o desenvolvedor que o escreveu já não estiver mais disponível.

Um olhar sobre o futuro

O Zope tem milhões de seguidores fanáticos – tudo por causa de suas características, sua flexibilidade e sua dependência de algo tão popular quanto o Python. Ao que parece, essa comunidade só tende a crescer. O uso do software para projetos em larga escala e em grandes empresas ajudou o programa a alcançar a maturidade que vemos hoje. O Zope X3 é um passo de gigante dado à frente

Figura 1: Esse formulário de edição foi gerado automaticamente por referência ao esquema de dados para o componente do conteúdo.

pelo pessoal que o mantém. Por outro lado, vai demorar um bocado até que o Zope X3 e seu novo paradigma ganhe as ruas, portanto devemos esperar ainda

vida longa ao Zope 2. Graças ao projeto Five [6], podemos tentar migrar do Zope 2 para o X3 sem pensar muito sobre as incompatibilidades da API.

Quadro 2: Instalação e configuração

Instalar o Zope no Linux é simples. Você vai precisar da atual versão 2.3.4 do Python com suporte a Zlib. Embora o Zope seja escrito em Python em sua maior parte, alguns módulos foram implementados em C por razões de velocidade. Será preciso compilar o aplicativo antes de instalar. O arquivo tar.gz tem um script `configure` que gera automaticamente um `Makefile` para o processo de compilação e instalação.

As bibliotecas do Zope normalmente são instaladas em `/usr/local/Zope-3.0.x`. Obviamente, você pode mudar essa localização editando o parâmetro `--prefix` no script `configure`. Para iniciar uma instância do servidor, você precisa primeiro criar uma árvore de diretório para ela. Ela é usada não somente para armazenar o banco de dados do objeto para a instância do Zope (ZODB) como também para armazenar bibliotecas anciãs exigidas especificamente para a instância. Claro que uma instalação do Zope pode usar diversas instâncias paralelas.

O script `mkzopeinstance` no diretório `bin` cria instâncias do programa. Se você especificar o caminho para a instância (cada instância fica em um diretório separado) e as credenciais para uma conta administrativa temporária, pode ativar uma instância digitando `runzope` (no diretório `bin` da instância). A configuração do servidor, as portas para HTTP e FTP e as várias opções de registro de eventos (*log*) estão disponíveis em `etc/zope.conf`. O arquivo de configuração usa o mesmo formato que o do servidor Apache. Por padrão, a instância do servidor web usará a porta 8080; o servidor de FTP usará a porta 8021.

Depois de iniciar o Zope digitando o comando `runzope`, pressione **[Ctrl]+[C]** para sair. Como isso é impraticável em uma aplicação rodando num servidor, é possível usar o script `zopectl` (também no diretório `bin` de cada instância) para controlar o servidor; o procedimento é similar ao do `apachectl` para o servidor web Apache.

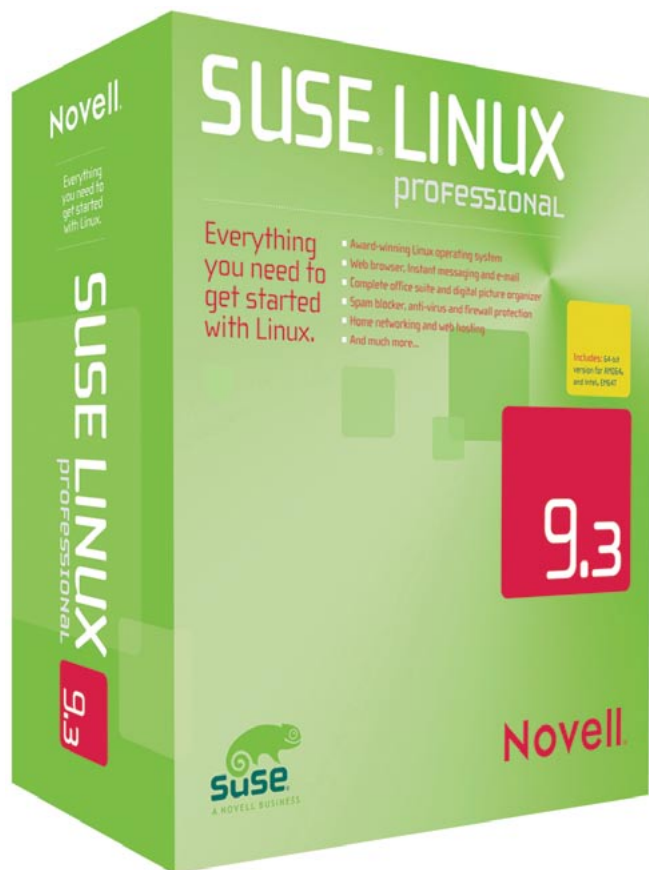
SOBRE O AUTOR

Philipp von Weitershausen estuda física em Dresden, na Alemanha. Também é um desenvolvedor freelance de software e consultor e membro do grupo que desenvolve o Zope 3. Seu livro, **Web Component Development with Zope 3** (Desenvolvimento de componentes para a web com o Zope 3) estará nas prateleiras europeias em meados de 2005.



INFORMAÇÕES

- [1] Plone: plone.org
- [2] Zope: zope.org
- [3] Silva: infrae.com/products/silva
- [4] CPS: www.nuxeo.org/cps
- [5] Zope X3: zope.org/Products/ZopeX3
- [6] Five: codespeak.net/z3/five



Camaleão posto à prova

9.3

A mais recente versão do SUSE LINUX Professional foi lançada em meados de abril. Como era de se esperar, a caixinha traz uma porção de programas antigos, alguns novos e um monte de módulos YaST aperfeiçoados, além de muita documentação. Continue a ler para descobrir o que mais mudou no SUSE 9.3 (além, é claro, das letras maiúsculas no nome).

POR MARCEL HILZINGER

À primeira vista, o *SUSE LINUX 9.3 Professional* pode não parecer que oferece muito no respeito a novas recursos. Excetuando-se as novas telas de inicialização (verdes, para variar), o KDE e o Gnome não parecem ter sofrido quaisquer mudanças radicais. Mas não se deixe enganar por essa primeira impressão. Os desenvolvedores do SUSE fizeram muitos e grandes progressos ao longo dos últimos seis meses. O SUSE LINUX 9.3 Professional inclui muitos novos recursos – e novos bugs a enfrentar.

Quadro 1: Números de versão

Programa	Versão
KDE	3.4
Gnome	2.10
Gimp	2.2.4
Adobe Acrobat Reader	7.0
Kernel	2.6.11.4
Glibc	2.3.4
Gcc	3.3.5
Samba	3.0.12
MySQL	4.1.10
PostgreSQL	8.0.1

Instalação

A instalação baseada no YaST (*Yet Another Setup Tool*) não mudou, exceto por dois detalhes: antes de iniciá-la, é preciso primeiro aceitar o acordo de licenciamento da Novell. E há uma outra surpresa à espera dos que preferem o particionamento manual: a Novell não oferece mais suporte ao *Journaling Filesystem* (JFS) da IBM por razões técnicas. Embora ainda seja possível montar partições JFS já existentes, não há como criar um novo sistema de arquivos JFS.

Nos três sistemas de nosso laboratório, o instalador identificou corretamente todo nosso hardware e o integrou automaticamente ao sistema em alguns casos; em outros casos, bastou um clique do mouse para pôr tudo para funcionar. No final da instalação, o YaST oferece uma atualização online e exibe as notas de lançamento da versão 9.3. As notas de lançamento alemãs são bastante concisas e não parecem completas. Se você realmente se interessa em saber com detalhes o que mudou, talvez prefira baixar a versão

mais recente da Internet ou ler as notas de lançamento originais em [/usr/share/doc/release-notes/](#).

Todos os usuários de computadores com placas de vídeo NVIDIA devem ficar atentos. Reinicie seu PC ao final da atualização online antes de instalar os drivers via YOU (*YaST Online Update*). Se não o fizer, o desktop talvez não consiga carregar em seguida a uma atualização do kernel.

No Desktop

A caixinha traz as mais recentes versões dos principais ambientes de trabalho: o KDE 3.4 e o Gnome 2.10. A Novell teria adorado acrescentar uma versão estável do *OpenOffice.org 2.0* – de fato, a tela de apresentação mostra esse número de versão. Na verdade, o que há ali é a versão de pré-lançamento, cujo número oficial é 1.9.79. A versão 2.2 do *Gimp* também faz parte do conjunto, instalada por padrão com o plugin para lidar com imagens no espaço de cor CMYK. O **quadro 1**, “Números de versão” apresenta a versão dos outros componentes principais.

A primeira mudança perceptível após a instalação é que os desenvolvedores do SUSE substituíram a tela de apresentação do KDE pelo tema de boot. Embora isso deixe a inicialização do sistema com uma aparência mais padronizada, não ajuda em nada a acelerar o processo. Se você prefere usar a tela de apresentação original do KDE, ou uma que você mesmo desenhou, é só escolher um tema da coleção encontrada no *Centro de Controle do KDE* em *Appearance and Themes | Splash Screen* (*Aparência e Temas | Tela de Apresentação*).

Também foi feita uma série de mudanças na aparência da área de trabalho do KDE (ver **figura 1**). Por exemplo: quando iniciado, o *Konqueror* abre uma janela com links para a lixeira, pastas, configurações, mídias removíveis de armazenamento e programas. Ao passar o mouse sobre um dos ícones do lado esquerdo do painel surge uma linda janela de informações para cada ícone. Infelizmente, o *Konqueror* também parece abrigar um ninho de baratas nojentas. Como documentado por muitos usuários da versão beta, o navegador ocasionalmente se recusa terminantemente a carregar uma página, queixando-se de que não pôde encontrar o servidor onde ela está hospedada. Já o *Firefox* (também instalado por padrão) consegue exibir a mesma URL sem quaisquer problemas. Tivemos esse

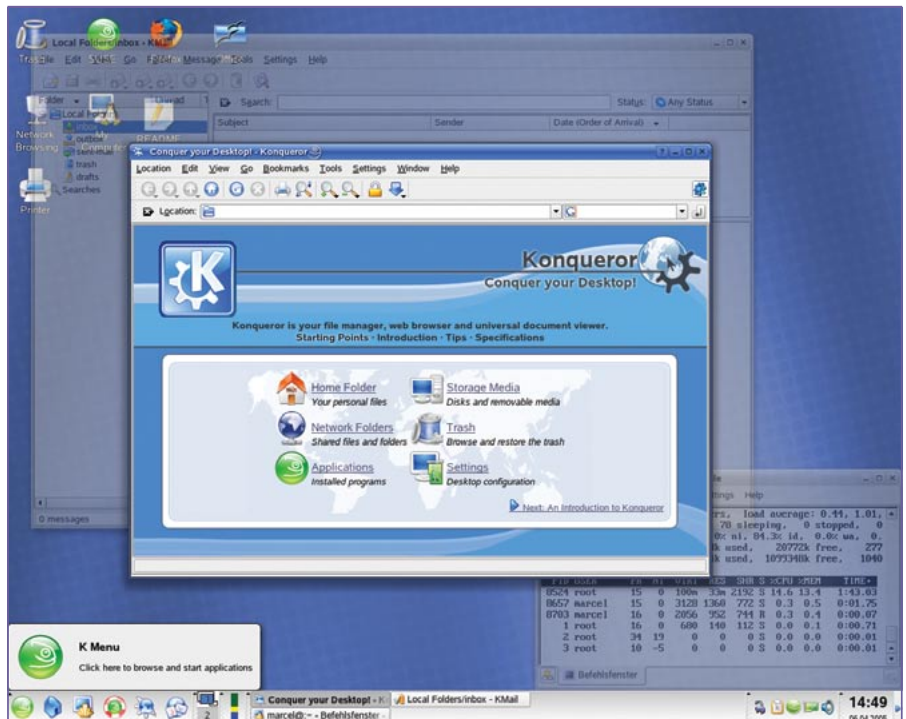


Figura 1: Se sua máquina for poderosa o bastante, você pode habilitar o novo look do SUSE com genuína transparência de janela.

mesmo problema em um dos sistemas de testes em nosso laboratório depois de iniciar a rede manualmente.

O SUSE finalmente disse adeus a seu próprio estilo de janela e o substituiu por uma versão ligeiramente modificada do estilo padrão do KDE 3.4, o *Plastik*. Desde que você tenha uma placa de vídeo que permita isso e poder de processamento para dar e vender, o KDE também oferece janelas transparentes, mas esse recurso vem desabilitado por padrão.

As mudanças mais visíveis podem ser atribuídas à nova versão 3.4 do KDE. Um novo recurso é o ícone *System* na área de trabalho. O ícone chama o endereço `media:/` e exibe os discos de sistema. O *Konqueror* mostra o nome de quaisquer partições rotuladas e o SUSE adiciona a capacidade ao nome se o disco ou dispositivo não tiver rótulo: por exemplo, *Média 23G* (notem

o erro na tradução da palavra em inglês *Media*. O correto seria *Mídia*) ou *Média Removível 15G* (veja a **figura 2**).

Debaixo dos panos

Se você não notou nenhuma mudança até agora, tente iniciar o *Amarok* e carregar sua própria coleção de músicas em MP3. Isso deve evidenciar bem algumas das mudanças. Ao clicar num arquivo MP3 no *Amarok*, nada acontece a princípio. Mas se você carregar a jukebox do KDE por meio do console, uma janela de terminal surge com uma importante mensagem: *Could not load mpeg-decoder*. Aparentemente a Novell decidiu não incluir um decodificador de MP3 no CD da distribuição no último minuto. Diferente da Red Hat, a Novell decidiu que os pacotes necessários estariam disponíveis apenas no *YaST Online Update*.

Por outro lado, a distribuição é a primeira a incluir drivers binários para as placas de vídeo *ATI Radeon* mais recentes. Todavia, apenas um dos dois pacotes ne-

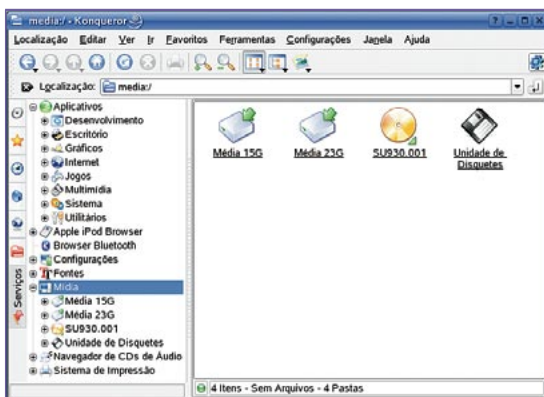


Figura 2: O SUSE LINUX 9.3 mostra o rótulo do disco, ou o tamanho e tipo, quando monta dispositivos de armazenamento.

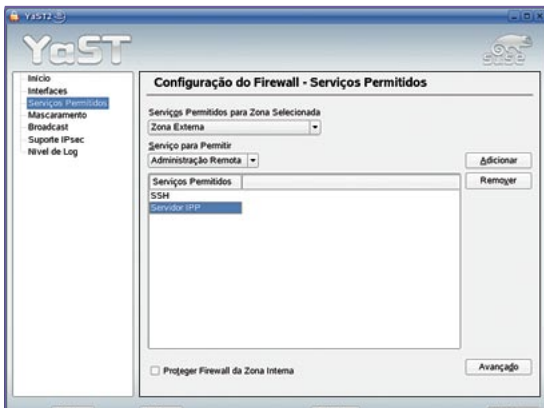


Figura 3: A Novell refez completamente o módulo de firewall do YaST.

cessários pode ser encontrado nos CDs: o módulo do kernel `km_fglrx`. Assim como com muitos outros pacotes, aparentemente não havia espaço nos CDs para o `fglrx`, mas ele está no DVD. Os drivers não são instalados automaticamente, nem exibidos para instalação.

Há também alguns novos drivers *Epkowa* para scanners Epson nos seguintes modelos: Epson Perfection 1250, 1250 Photo, 1260, 1260 Photo, 3170 Photo, 4180 Photo, GT-7200U, GT-7300U, GT-9400 e GT-F600. Para configurar um desses scanners, rode o módulo *Scanner* do YaST e inicie o programa *Iscan*.

A Novell usou o HAL e a arquitetura D-Bus no SUSE 9.3. Dispositivos que podem ser detectados pelo subsistema *Hotplug*, como chaveiros USB, discos rígidos ex-

ternos e gravadores de CD e DVD se beneficiarão muito disso. Nesse quesito, o SUSE 9.3 foi enormemente aperfeiçoado em comparação com suas versões anteriores.

Houve mudanças em diversos módulos do YaST. Por exemplo, três abas distintas para *User data*, *Details* e *Password settings* tornam a criação e a modificação de contas de usuário muito mais fáceis.

A Novell também retrabalhou completamente o módulo do Firewall (ver **figura 3**). Agora você pode definir diversas zonas e habilitar, por exemplo, o *IPSec*. Há também uma lista de serviços que permite aos administradores ativar (ou desativar) qualquer um deles com um simples clique.

O módulo da impressora também é afetado pelas configurações de firewall. Como sempre, o SUSE configura a máquina como cliente CUPS por padrão na instalação, mas o 9.3 deixa de permitir acesso à porta necessária na configuração padrão. É bem fácil passar por cima desse detalhe durante a instalação; assim, tenha a certeza de abrir a caixa de diálogo da impressora ou do firewall ao terminar e acrescente *IPP-Client* à lista de serviços permitidos (**figura 4**).

A Novell também introduziu uma porção de mudanças para dispositivos de rede. Agora esses dispositivos podem ser habilitados automaticamente, via *Hotplug*, no momento da inicialização, bastando ligar um cabo de rede. Há também uma caixa em que os administradores do sistema podem permitir que os usuários habilitem serviços de rede.

Por último, mas não menos importante, há também dois novos módulos YaST. *Software*

| *Media Check* (verificação de mídias) permite procurar erros em CDs e DVDs de instalação do sistema. Esse recurso também está disponível durante a instalação, mas apenas em modo texto. Para usá-lo, basta sair da instalação gráfica e selecionar *Check Installation CD-ROM/DVD* no menu principal do *Linuxrc*.

O item *Installation into Directory for XEN* (instalação em um diretório para o XEN) permite a fácil criação de instalações virtuais do XEN. O XEN é uma solução de código aberto que permite que múltiplas instâncias virtuais do Linux rodem num único sistema hospedeiro, também Linux. Para mais informações sobre o XEN, veja [2].

KDE versus Gnome

O KDE ainda é a área de trabalho padrão do SUSE. Porém, uma visão geral mostrada antes da instalação dá a opção de selecionar um sistema Gnome como ambiente de trabalho. Nesta versão do SUSE, a Novell continua a trabalhar na integração entre o KDE e o Gnome. Diferente da Red Hat, a Novell não está muito preocupada com a padronização da aparência, concentrando-se em vez disso na usabilidade. Agora ambos os ambientes têm uma estrutura de menu mais ou menos idêntica, sendo a única diferença o fato do Gnome usar uma ordem estritamente alfabética enquanto o KDE simplesmente ordena os itens do menu, mas não suas categorias.

A Novell também acertou em cheio com a área de trabalho comum no SUSE 9.3. Os itens do KDE e do Gnome não aparecem como forasteiros ou duplicados no outro desktop. Os ícones *Susewatcher* e *Suseplugger* integram-se agora lindamente ao painel do Gnome. O Gnome passou por um teste com uma pasta de rede que criamos no KDE sem derramar uma gota de suor. E o Nautilus consegue abrir pastas do *Samba* sem qualquer problema após pedir seu nome de usuário e sua senha.

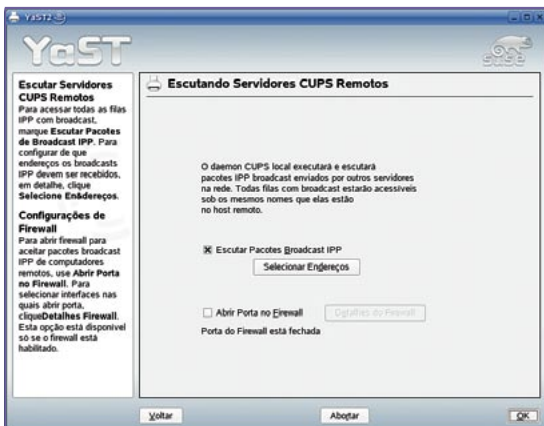


Figura 4: No YaST, crescente o *IPP-Client* à lista de serviços permitidos.

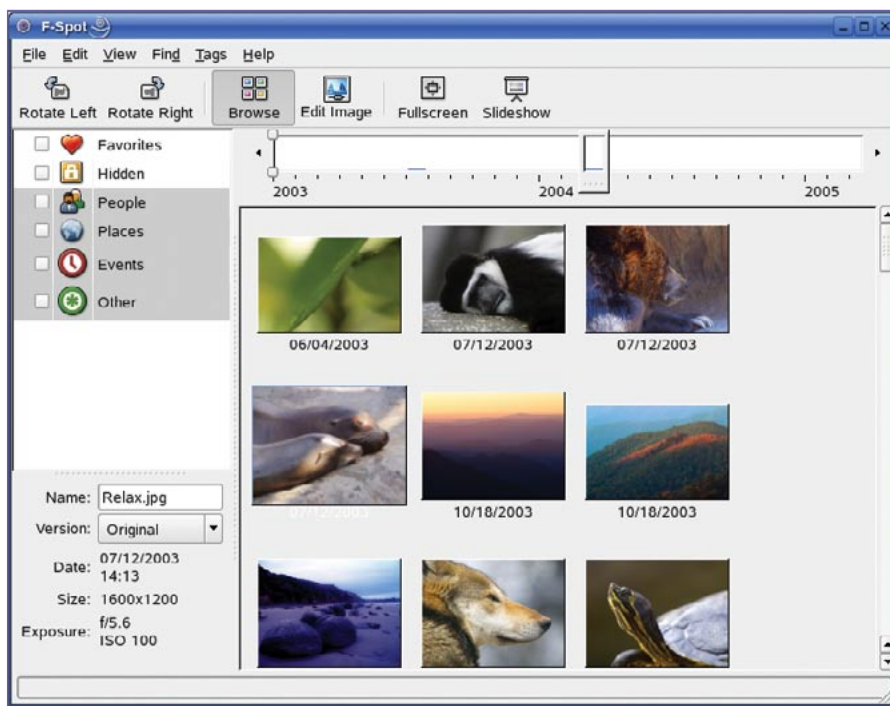


Figura 5: O F-Spot traz ao SUSE LINUX 9.3 um rápido visualizador de imagens e gerenciador de fotos.

A integração do OpenOffice.org ao GTK+ e ao QT é na maior parte bem-feita, embora aparentemente haja um pequeno bug na versão GTK do OpenOffice.org. Em dois dos três sistemas de testes em nosso laboratório, o programa não tinha o *anti-aliasing* de fontes na primeira vez em que o rodamos no Gnome. Mas esse problema desapareceu assim que instalamos uma nova fonte para a interface do usuário (guia *Interface User*) pelo menu *Tools | Options | Fonts (Ferramentas | Opções | Fontes)* e reiniciamos o OpenOffice.org. Não fomos capazes de reproduzir o bug depois disso.

O mecanismo de busca da área de trabalho, *Beagle*, só funcionou no Gnome em nosso primeiro teste. A razão para isso é que a ferramenta é baseada no sistema D-Bus e apenas o Gnome oferece todo o conjunto de recursos. O Beagle e o front-end que o acompanha, *best*, são uma versão alfa, por enquanto. Mesmo assim, a ferramenta, que foi escrita em Mono e, grosso modo, é uma mistura avançada de *locate* e *grep*, se mostrou

bastante confiável e rápida em nossos testes iniciais. Como o serviço é desabilitado por padrão, é preciso antes de tudo criar um arquivo `.runbeagle` vazio em seu diretório pessoal antes de começar as buscas. Há um plugin à parte para buscas no Firefox.

Os usuários do KDE, para evitar isso, podem iniciar o mecanismo de busca uma vez com o Gnome; isso fará com que seja possível rodá-lo na área de trabalho do KDE depois. Para isso, execute o comando `beagled` num shell e rode então o front-end gráfico com o comando `best`. A partir daí, é só pressionar a tecla **[F12]** a qualquer momento para usar a ferramenta de busca.

Após uma instalação padrão, outro programa em Mono reside sob *Graphics | Photography (Gráfico | Fotografia)*: o visualizador *F-Spot* (ver figura 5). Assim como a ferramenta *Digikam* do KDE, o F-Spot busca numa pasta livremente configurável e nas subpastas abaixo dela por imagens quando é iniciado e mostra os resultados como um álbum.

Linux fácil de usar

A Linux New Media do Brasil está trazendo para o mercado nacional a *Easy Linux*, um novo conceito em publicações para o iniciante no mundo da informática. Baseada no SUSE LINUX 9.3, e desenvolvida em parceria com a Novell na Alemanha, a primeira edição é composta por quatro CDs e um guia de instalação passo-a-passo com 32 páginas, escrito em uma linguagem fácil de entender, o que possibilita a qualquer um instalar o sistema. Além do guia de instalação, o usuário também poderá tirar suas dúvidas usando o suporte técnico gratuito via Internet.

O sistema, em português do Brasil, tem uma interface amigável e fácil de usar e é composto pelo ambiente de trabalho KDE 3.4, o conjunto de aplicativos de escritório *OpenOffice.org 2.0*, compatível com o *Microsoft Office*, o navegador *Mozilla Firefox*, o cliente de e-mail/organizador pessoal *Kontact*, além de muitos outros utilitários, aplicativos para visualização de arquivos multimídia, CDs, DVDs e até mesmo jogos.

O produto visa atender os usuários domésticos que querem mais segurança e estabilidade em seus computadores, sem ter de se preocupar em adquirir e manter anti-vírus, removedores de *spyware* e *firewalls* pessoais, e pequenas empresas que desejam legalizar seu parque de software sem os altos custos de licenciamento do software proprietário. Mais informações podem ser obtidas no site oficial, em www.revistaeasylinux.com.br



Diferente do Digikam, o F-Spot não organiza as imagens em diretórios individuais, mas exibe toda sua coleção. Você pode usar uma linha do tempo ou atribuições de categoria para fins de seleção das fotografias. Se necessário, o programa pode puxar imagens diretamente de sua câmara digital. O F-Spot é rápido, mas não pode exibir imagens no formato PNG, o que influi em sua usabilidade como um visualizador para uso geral, embora esse não seja seu verdadeiro propósito.

Vale a pena atualizar?

Se você já tiver o SUSE LINUX 9.x ou algum sistema igualmente recente, a atualização só vale a pena se você estiver infeliz com sua configuração atual. Elaboramos uma listinha de prós e contras para ajudar você a se decidir.

O OpenOffice.org "2.0" (que, como já dissemos, é na verdade a versão de pré-lançamento 1.9.79) parece limpo e fresco, embora não tenham havido aperfeiçoamentos drásticos no tempo que os componentes individuais demoram para carregar – dois ou três segundos a menos é o máximo que se pode esperar. O conjunto de aplicativos de escritório travou algumas vezes durante nossos testes, mas conseguiu resgatar sem problemas os documentos que não estavam salvos. O novo módulo de banco de dados facilita a vida para os usuários do Access, mas não pode competir com a riqueza de recursos oferecida pelo Knoda [3].

Se você usa regularmente mídia removível, a atualização é sem sombra de dúvida uma boa idéia. A Novell resolveu os problemas que assolavam os chavesiros USB na versão 9.2. Ao montar uma partição, o SUSE Linux não usa mais as IDs de dispositivo ridiculamente longas, mas – se existir – o rótulo das partições. Em caso de mídias externas sem rótulo, o SUSE usa o ponto de montagem `/media/usbdisk`. O sistema monta corretamente chavesiros USB sem tabelas de partição.

Trocar o CD ou o DVD que estão no drive por outros é bastante incômodo, pois o ponto de montagem recebe o nome do rótulo da mídia. Isso faz com que o simples comando `cd /media/dvd/` aponte para um buraco negro.

A montagem de grandes discos externos com capacidade de centenas de gigabytes ainda é uma experiência frustrante. Para permitir que mídias sejam desmontadas a qualquer momento sem rodar o comando `umount`, o SUSE LINUX usa a opção `sync` para sistemas de arquivo. Porém, isso implica em um excesso de atividade em discos que usam o sistema de arquivos ReiserFS: o disco sincroniza continuamente e por isso roda bem mais devagar. Um jeito de evitar isso seria montar o disco manualmente. A melhor idéia é usar o ponto de montagem escolhido pelo SUSE para montar o disco. Não é necessário desmontar automaticamente partições montadas para isso, nem mesmo temporariamente.

Usuários de laptops certamente se beneficiarão da atualização. As funções de gerenciamento de energia (ACPI, veja mais sobre o assunto em [5] e [6]) no SUSE LINUX 9.3 foram enormemente aperfeiçoadas. O *suspend-to-disk* agora funciona em máquinas com drivers NVIDIA e leva cerca de 20 segundos num Dell Inspiron 8600, por exemplo. Com o 9.3, esses computadores suportam até mesmo o modo *standby* planejado pelo fabricante. De acordo com a Novell, criar uma partição Ext2 ou Ext3 distinta para o diretório `/boot` aumenta a velocidade do processo de suspensão. Entretanto, após seguir esse conselho, não notamos qualquer diferença em nossos testes.

Voltar à ativa após um *suspend-to-ram* ainda é questão de pura sorte para a maioria das máquinas. Felizmente, o SUSE LINUX 9.3 tem algumas ferramentas que podem tirar até mesmo a mais obstinada das máquinas da hibernação. O arquivo `/usr/share/doc/packages/`

`powersave/README.suspend2ram`, que também tem uma lista dos laptops testados, traz mais informações.

Conclusão

O SUSE LINUX 9.3 da Novell é uma distribuição Linux robusta e rica em recursos. À primeira vista, a nova versão não parece ter mudado grande coisa desde a 9.2, mas houve muito trabalho por baixo dos panos. A distribuição é definitivamente uma boa escolha para novos usuários. Uma atualização é recomendada para laptops e computadores que usam mídias removíveis com frequência.

Não houve grandes surpresas quando pusemos o SUSE 9.3 à prova em nosso laboratório. Se você tiver problemas para instalar o sistema, aconselhamos ler o artigo *Known issues and special features in SUSE LINUX 9.3* no banco de dados do suporte do SUSE [4]. ■

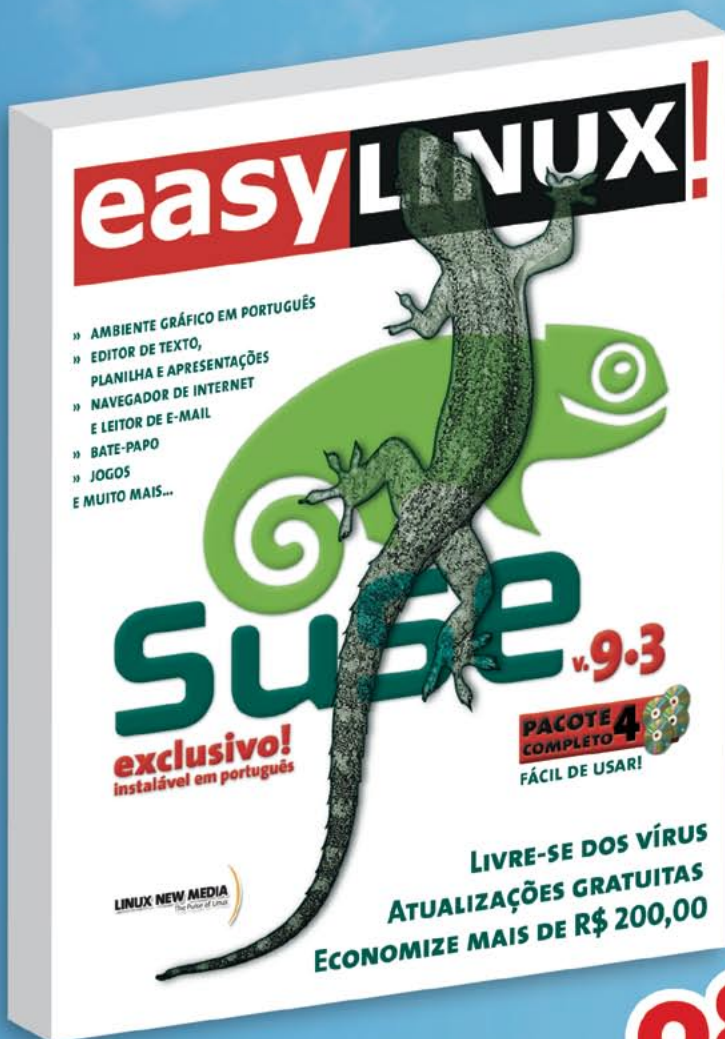
SUSE LINUX 9.3 Professional

Preço	£ 64.95, US\$ 99.95, EUR 89.95
Suporte	90 dias após a instalação
Websites	www.suse-brasil.com.br shop.novell.com www.novell.com/products/linuxprofessional

INFORMAÇÕES

- [1] Site oficial: www.novell.com/products/linuxprofessional
- [2] Marco SInhoreli, *Multiplicando Pingüins*. Linux Magazine Brasil, 9ª. Edição, página 40.
- [3] Marcel Hilzinger, *Organizando seus dados*. Linux Magazine Brasil, 8a. Edição, página 61 e *Hmm... casa arrumada, hein?*. Linux Magazine Brasil, 9ª. Edição, página 48.
- [4] Banco de dados do suporte do SUSE: portal.suse.com/sdb/en/index.html
- [5] Simon Rutishauser, *Colocando seu laptop para dormir*. Linux Magazine Brasil, 5ª. edição, página 64.
- [6] Simon Rutishauser, *Sono Profundo*. Linux Magazine Brasil, 5ª. Edição, página 68.

Sinta-se livre!



easyLINUX!

O JEITO MAIS FÁCIL DE COMEÇAR

- » Livre-se agora dos vírus e dos travamentos constantes com um sistema robusto e fácil de instalar e usar. Acompanha 4 CDs com utilitários, jogos e aplicativos atualizados e seguros.
- » Abra, edite e salve arquivos nos formatos dos aplicativos de escritório da Microsoft (Word, Excel e Powerpoint) ou da Adobe (Photoshop, PDF).
- » Navegue na Internet à toda velocidade sem se preocupar com propagandas e ataques, livre-se automaticamente do SPAM.

COMPRE AGORA PELO TELEFONE E RECEBA EM CASA

0800-702-5401

WWW.REVISTAEASYLINUX.COM.BR

As novidades do Gnome 2.10

Seguindo as pegadas

O novo ambiente gráfico Gnome traz muito mais aplicativos e applets. E nos bastidores o Gnome 2.10 está ainda mais robusto e inovador.

POR CHRISTIAN MEYER

Desde a versão 2.0, o Gnome recebeu tantas críticas quanto aplausos. Há apenas dois anos e meio, ninguém teria achado possível que um ambiente gráfico com um ciclo de lançamento de seis meses manteria os altos padrões de qualidade que vêm sendo mantidos pelo Gnome. Apesar da constante integração de novas funções, as bibliotecas básicas continuam a ser compatíveis com software mais antigo, um dos argumentos que ajudou a convencer algumas das principais empresas de TI, como a Red Hat, a Novell e a Sun, a usar o Gnome como base para suas linhas de produtos.

A nova versão 2.10 do Gnome já está disponível ao público para download desde 9 de março. Este artigo trata das mudanças mais importantes introduzidas pelo Gnome 2.10, muitas das quais não estão nos bastidores, mas sob as luzes da ribalta.

Mudanças à flor da pele

Ao iniciar a área de trabalho, você não pode deixar de notar a nova estrutura do menu no painel do Gnome. Enquanto as versões anteriores tinham dois itens, *Applications* e *Actions*, o Gnome 2.10 apresenta uma estrutura completamente nova com três categorias: *Applications* (aplicativos), *Places* (lugares) e *System* (sistema). Como a nova interface ainda não está completamente traduzida para diversos idiomas, apresentaremos os menus e seus itens em inglês.

A categoria *Application* ainda engloba todo o conjunto dos programas do Gnome (algo como o “menu iniciar” do Windows, ou o “menu da maçã” no Mac OS clássico), mas agora tem também o item *Run program...* (executar programa), que era anteriormente atribuído à categoria *Actions*.

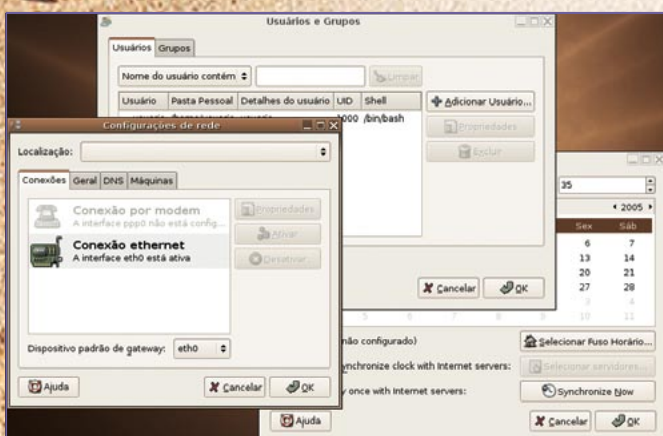


Figura 1: As Ferramentas de Sistema do Gnome têm uma interface apropriada para o gerenciamento do ambiente.



Figura 2: O versátil reprodutor de arquivos multimídia Totem é simples de configurar e usar.

O item *Places* leva a diretórios utilizados com frequência, incluindo diretórios pessoais, favoritos do gerenciador de arquivos, quaisquer mídias removíveis que estejam montadas no momento, servidores de rede e os últimos documentos abertos. Dali é possível conectar-se a um servidor e buscar arquivos armazenados local ou remotamente.

Por fim, *System* tem um submenu de configuração da área de trabalho e opções para tirar *screenshots*, trancar a tela e finalizar a sessão. O submenu de administração leva às Ferramentas de Sistema do GNOME (*GNOME System Tools*), onde se realizam todas as tarefas de gerenciamento do sistema.

Configuração fácil

Os administradores podem usar as *GNOME System Tools* (Ferramentas de Sistema do GNOME) para gerenciamento simples de usuários ou para ajustar a data e a hora, inclusive ajustar a hora automaticamente via Internet, por meio de um servidor NTP. A configuração de rede também é sopa no mel. Os usuários de laptops vão gostar de poder criar perfis ou localizações. Dependendo de onde você esteja, em casa ou no trabalho, por exemplo, é só selecionar o perfil apropriado àquele local. As *GNOME System Tools* permitem que os administradores

configurem modems, dispositivos ISDN e redes sem fio. É possível definir compartilhamentos NFS ou Windows/Samba, permitindo que outros usuários os acessem. A figura 1 mostra as Ferramentas de Sistema do GNOME.

Novos programas

O GNOME 2.10 possui agora os populares aplicativos de multimídia Totem (figura 2) e o Sound Juicer. O Totem é uma ferramenta multifuncional que se baseia por padrão no *framework Gstreamer* [1], embora possa usar bibliotecas do Xine se você preferir; essa pode ser a melhor opção em alguns casos.

O Totem pode tocar tanto arquivos de áudio como de vídeo. O programa é de fácil configuração, característica essa que tanto os recém-chegados quanto os mais experientes saberão apreciar. Entre outras coisas, pode-se configurar o sistema de áudio (o que é interessante para sistemas *surround* 4.1/5.1) e os efeitos visuais ao tocar arquivos de áudio.

O “ripador” de CDs Sound Juicer converte sua coleção de CDs num formato mais conveniente para armazenamento digital, como MP3, Ogg Vorbis ou FLAC. O programa funciona de forma mais ou menos autônoma, incluindo a atribuição automática de título, artista, álbum e gênero às faixas criadas.

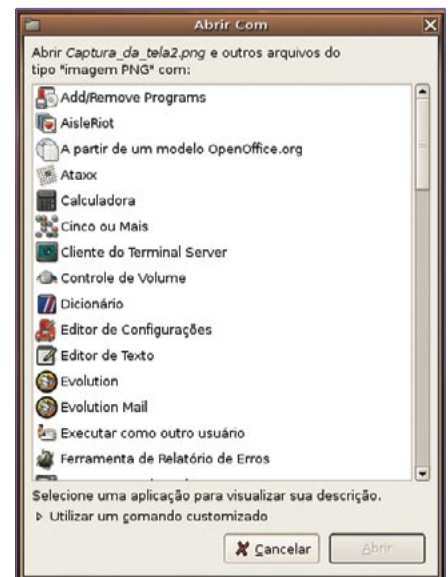


Figura 3: A nova caixa de diálogo Abrir com... oferece aos novatos um modo mais intuitivo de associar aplicativos no Linux.

Melhorias

O Nautilus, o gerenciador de arquivos do GNOME, passou por muitas atualizações práticas. A substituição de alguns componentes do sistema fez maravilhas pelo desempenho do Nautilus; eis aí algo que você vai notar assim que iniciar o gerenciador de arquivos e ao usá-lo. Além disso, o Nautilus renomeia novos diretórios imediatamente – nas versões anteriores, os usuários eram obrigados a fazê-lo manualmente. O atalho de teclado **[Ctrl] + [L]** leva diretamente a um diretório específico. A conectividade de rede também foi aprimorada, assim como a função de gravação de CDs integrada e a caixa de diálogo *Abrir com...*, que juntos tornarão a vida muito melhor para os calouros do Linux (ver figura 3).

O gerenciador de janelas Metacity não mais desvia o foco da janela em que o usuário está trabalhando. Esse comportamento irritante afetava a produtividade dos usuários do sistema nas versões anteriores.

A biblioteca GTK+ traz para os desenvolvedores um recurso de autocompletar código que pode ser usado por qualquer programa em GTK+. Na visualização em

árvore ou como lista, basta digitar as primeiras letras da palavra que se procura e a seleção pulará imediatamente para a primeira ocorrência. O próprio Nautilus não tem esse recurso, mas possui algo similar. Parece bastante provável que o Gnome 2.12 terá este mimo aplicado a todos os seus componentes.

E por fim, mas não por isso menos importante, há a caixa de diálogo *Desktop Background Preferences* (figura 4); o “arrastar e soltar” com o Mozilla foi também aperfeiçoado. Além disso, os desenvolvedores removeram uma porção de pequenos bugs de todo o ambiente gráfico.



Figura 4: A caixa de diálogo *Desktop Background Preferences* em suas novas vestes.

Navegadores web e email

O navegador web *Epiphany*, baseado no interpretador HTML *Gecko* (o mesmo usado no *Mozilla*), agora é completamente compatível com as versões atuais do *Mozilla* e do *Firefox* e tem um modo de tela cheia otimizado. Assim como o *Firefox*, o *Epiphany* agora exibe um ícone para que os usuários saibam se estão acessando uma página segura (via *https*). As extensões permitem que você aumente a lista de recursos do *Epiphany*. Versões anteriores do navegador carregavam todas as extensões ao iniciar o programa e consumiam grandes quantidades de memória ao fazê-lo. Agora há um geren-

ciador de extensões para uma integração e administração mais conveniente dos módulos necessários.

A solução groupware *Evolution* combina gerenciamento de email, contatos e lista de tarefas, função de calendário e muitas outras coisas. Por funcionar tanto com o *Novell Groupwise* quanto com o *Microsoft Exchange*, ele se integra suavemente a ambos os ambientes; além disso, funciona também no *Scalix* [2] e no *Sun Java Enterprise System Calendar Server*. A Novell atualmente vem investindo muito na continuidade do desenvolvimento do *Evolution*;

O projeto *Hula* da Novell [3], em particular, causou bastante rebuliço recentemente. Fique de olho nesta coluna para conhecer mais detalhes sobre um conector muito em breve.

Os usuários vão gostar muito de saber que agora o suporte para modo offline está mais suave que em versões anteriores. Arquivos podem ser anexados a eventos e o *Evolution* pode ser melhorado com extensões, que são ativadas e desativadas numa tela de administração, de maneira

similar às extensões do *Epiphany*.

Programas

O *GnomeMeeting* está ainda mais bem integrado ao sistema; ele compartilha o livro de endereços do *Evolution*, o que quer dizer que agora os usuários só precisam digitar uma vez as informações dos contatos para que elas fiquem disponíveis globalmente. Se você tiver o *Zeroconf* [4] instalado, o *GnomeMeeting* localizará automaticamente outros usuários do *GnomeMeeting* numa rede.

O editor de texto *gedit* não apenas oferece destaque de sintaxe, como também localiza e realça pares de parênteses em

código-fonte (figura 5). Isso é ótimo para os programadores acostumados com esse recurso em sua experiência com o *Emacs*, o *Vim* e outros ambientes de desenvolvimento. Scripts em *Bash* e *Tcl* são, agora, reconhecidos. Como é o caso com muitos programas, o tempo necessário para o *gedit* iniciar reduziu-se drasticamente.

A mais recente versão do gerenciador de arquivos *File Roller* agora permite o uso de arquivos do tipo AR (tipo padrão de compressão do Debian), pacotes Debian (.deb), arquivos RAR protegidos por senha e arquivos 7 ZIP. Ele também reflete o comportamento dos cliques configurado no *Nautilus*. Se você optar pelo duplo clique no gerenciador de arquivos, o *File Roller* fará a mesma coisa.

O dicionário do Gnome agora tem um corretor ortográfico. A vantagem dele é que encontra as palavras mesmo quando você não tem certeza de como soletrá-las. Assim como nos *wikis* (veja nossas matérias de capa), os textos agora estão vinculados a outros dicionários na web e é possível imprimi-los.

O visualizador do registros (*logs*) do sistema era uma ferramenta bastante negligenciada nas versões anteriores. De fato, algumas pessoas eram a favor de removê-lo do núcleo do Gnome. O Gnome 2.10 vê a introdução de alguns interessantes novos recursos para ela. Por exemplo, é possível agora monitorar múltiplos arquivos, incluindo arquivos

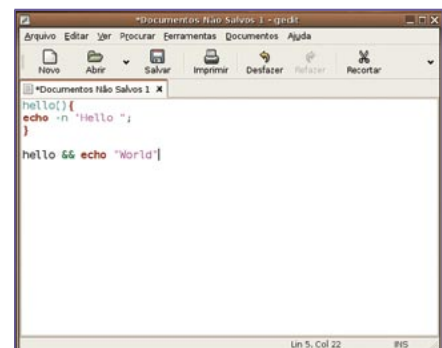


Figura 5: Além de realçar a sintaxe, agora o *gedit* destaca pares de parênteses.

comprimidos ou não-locais (em drives de rede), com o visualizador. Ele tem também uma eficiente barra de buscas e permite que os usuários copiem dados para a área de transferência.

A ferramenta de formatação de discos também foi completamente retrabalhada. Se antes só era capaz de formatar disquetes, na nova versão funciona com qualquer mídia removível.

Painel e applets do Gnome

No que tange ao elemento central de controle, os aperfeiçoamentos feitos no painel incluem suporte para temas de ícones e *applets* com suporte a imagens vetoriais no formato SVG (*Scalable Vector Graphics*, da Adobe). A caixa de diálogo *Run program...* (*Executar programa...*) está bem mais rápida e os *applets* nos painéis transparentes agora podem ter fundo realmente transparente.

O applet de controle de montagem de discos não enlouquece mais os usuários com configuração complexa e jargão técnico. Ícones de discos escondidos atrás de janelas da área de trabalho deixaram de ser um problema. O applet, que é visível no painel, exibe todos os discos montados e permite até mesmo que os administradores os desmontem.

O applet de clima agora engloba muito mais localidades e informa os usuários sobre o tempo no lugar escolhido, incluindo as horas da alvorada e do pôr-do-sol (recurso disponível apenas para as grandes cidades).

O applet *sticky label* agora lembra em qual área de trabalho e monitor você grudou um lembrete. A lista completa de novos *applets* fornecidos com o Gnome 2.10 também inclui:

→ Um applet de modem, que se integra perfeitamente à interface de administração do sistema.

→ Uma lixeira que permite arrastar e soltar arquivos para jogá-los fora da maneira usual. Um clique no ícone abre a lixeira e mostra os arquivos e diretórios ali colocados.

→ Um mostrador da velocidade da CPU, útil para mostrar a velocidade em laptops equipados com tecnologia AMD PowerNow! ou Intel SpeedStep (**figura 6**). Ali também é possível selecionar manualmente uma velocidade, se o ajuste automático não estiver habilitado.

Aos *applets* existentes foi adicionado um mixer de áudio. Como o restante do ambiente, ele usa a estrutura Gstreamer. A caixa de configuração é mais limpa e traz as abas *Playback*, *Capture*, *Switches* e



Figura 6: Um novo applet permite ajustar a velocidade do processador de seu notebook.

**COMPLETE
A SUA COLEÇÃO!**
COMPRE AS EDIÇÕES ANTERIORES
DA SUA LINUX MAGAZINE,
POR TELEFONE OU PELO NOSSO SITE.

LINUX NEW MEDIA
The Pulse of Linux

0800-702-5401
www.linuxnewmedia.com.br



Options (figura 7). Se o sistema de som ALSA estiver instalado, você pode alternar entre OSS (o sistema de som antigo e já obsoleto) e ALSA (que se tornou o padrão a partir do kernel 2.6). Isso evita aquele amontoado de controles deslizantes e botões, uma praga comum em outros ambientes.

A nova versão do Gnome livrou-se de três applets, que foram substituídos ou abandonados. São esses o applet de CD Player; o applet de WLAN, já que suas funções foram integradas ao applet de monitoramento de redes desde o Gnome 2.8; e o alerta de recebimento de email, pois seu mantenedor não tem tempo para mexer nele.

Jogos e diversões

Agora os usuários podem desafiar seus vizinhos da rede local para um jogo de *Nibbles* (que usa a mesma tecnologia do GnomeMeeting para descobrir parceiros). Embora o jogo seja muito simples, está mais divertido do que nunca.

O jogo *Same Gnome* ganhou novos gráficos e mais tamanhos de tabuleiro, além de permitir desfazer as jogadas impen-sadas. Alguns bugs que interrompiam o fluxo do jogo também foram removidos.

Futuro

Um tópico que vem sendo objeto de inflamação discussão no momento é a quantidade de memória exigida por aplicativos baseados em GTK+. Os desenvolvedores estão trabalhando duro na resolução desse problema. Você pode conferir o progresso deles em [5].

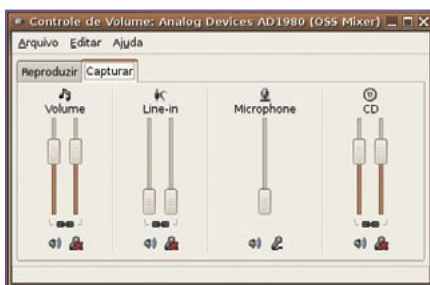


Figura 7: O applet mixer parece realmente limpo, já que se livrou dos elementos supérfluos.

O Gnome 2.12, que deve ser lançado em meados de setembro, já está sendo desenvolvido em paralelo. Não podemos dizer com certeza o que o 2.12 trará no quesito “novos recursos”, mas vamos olhar dois exemplos só para deixar você com água na boca.

Um editor de menu foi por muito tempo o maior sonho de consumo para muitos usuários. Embora o Gnome 2.10 honre as especificações do Freedesktop para menus [6], não oferece aos usuários uma ferramenta conveniente para administrá-los. O Gnome 2.12 parece pronto para mudar esse quadro.

As coisas estão também começando a se agitar no mundo encantado dos ambientes gráficos. O *Looking Glass* da Sun [7] demonstra que isso não é apenas um truque publicitário. Um Gnome em 3D poderia ser baseado no *Luminocity*, um híbrido de gerenciador de janelas e de composição. Para ver com seus próprios olhos, dê uma olhada nos impressionantes vídeos do *Luminocity* [8] para conhecer mais detalhes sobre este fascinante projeto.

Se alguns dos recursos do *Luminocity* estiverem estáveis na época do lançamento do Gnome 2.12, você já pode se preparar para vê-los no *Metacity*, o gerenciador de janelas padrão do Gnome. Porém, o *Luminocity* deve ser visto como um ambiente de teste por enquanto; não está pronto para máquinas de produção. Se você se interessou e quer experimentar mesmo assim, veja em [9] um pequeno guia de compilação do *Luminocity*.

Conclusão

A versão 2.0 do Gnome mostra o ambiente de trabalho do pezinho caminhando a passos largos em direção ao objetivo de ser o software mais intuitivo de sua categoria. A configuração é simples e o ambiente, produtivo. Muitos dos recursos que estavam faltando nas versões mais antigas foram adicionados ou completados nos últimos meses.

Agarrando um gnomo

A conferência de desenvolvimento do Gnome, GUADEC [10], marcada para ocorrer em Stuttgart, Alemanha (por ironia, a terra do KDE), no fim de maio, promete trazer aos participantes uma boa compreensão e uma visão geral do futuro do Gnome. A lista VIP para a GUADEC inclui Mark Shuttleworth (fundador da *Canonical* e do projeto *Ubuntu Linux*) e Miguel de Icaza (Atualmente funcionário da Novell e fundador da *Ximian* e do projeto Gnome).

O Gnome 2.12 continuará a seguir essa trilha e tentará aparar algumas arestas. A biblioteca GTK+ 2.8 promete dar um enorme impulso na velocidade, que não faz muito bonito em nenhum dos programas baseados em GTK+ até agora. Porém, não há como afirmar se essa versão será a base do próximo Gnome. ■

INFORMAÇÕES

- [1] GStreamer: www.gstreamer.net
- [2] Scalix: www.scalix.com
- [3] Projeto Hula: hula-project.org
- [4] Zeroconf: zeroconf.sourceforge.net
- [5] Informações sobre o uso de memória no GTK+: live.gnome.org/MemoryReduction
- [6] Especificações sobre menus do Freedesktop: freedesktop.org/wiki/Standards_2fmenu_2dspec
- [7] Looking Glass da Sun: www.sun.com/software/looking_glass/
- [8] Vídeos do Luminocity: www.gnome.org/~seth/blog/xshots
- [9] Como compilar o Luminocity (em inglês): live.gnome.org/Luminocity
- [10] GUADEC 2005: 2005.guadec.org

SOBRE O AUTOR

Christian Meyer já trabalha com o Gnome há cinco anos e é o presidente da *Gnome Deutschland e.V.* (uma organização sem fins lucrativos que apóia e promove o uso do Gnome na Alemanha), fundada em outubro de 2004. Christian pode ser contatado, em alemão ou inglês, através do email chrisime@gnome.org.

Usando uma conta de email como sistema de arquivos remoto

Dados pra viagem

Por que deixar uma conta de e-mail de um gigabyte (dependendo da conta, dois) ser desperdiçada?

O GmailFS permite que você use uma conta do Gmail como um sistema de arquivos móvel.

POR FABRIZIO CIACCHI

O Gmail [1], sistema de *webmail* do Google, foi festejado como o novo paradigma do correio eletrônico via Internet. Uma conta do Gmail vem com um espaço para guardar mensagens que pode chegar aos 2 GB. De acordo com a empresa, possuir uma conta no Gmail implica em que o usuário não terá que jogar nenhuma de suas mensagens fora, nunca!

Entretanto – e como era de se esperar – oferecer gratuitamente uma grande quantidade de espaço de armazenamento a um usuário de Linux abre as portas da criatividade em certas cabecinhas privilegiadas, levando à imediata inovação. Logo depois da distribuição das primeiras contas de Gmail, os linuxistas encontraram um novo uso para esse espaço, completamente diferente da intenção original do Google. O GmailFS, criado por Richard Jones [2], usa os gigabytes de sua conta no Gmail como se fosse um diretório compartilhado em outra máquina da rede.

O que é, então, o GmailFS?

O GmailFS aparece para o usuário como um sistema de arquivos comum. De acordo com seu criador, “o GmailFS permite que

a maioria das operações com arquivos sejam feitas, como ler, gravar, abrir, fechar, conferir o estado, criar um link simbólico (*symlink*) truncar e renomear”. Trocando em miúdos: é possível usar qualquer dos comandos do Linux com o GmailFS, como `cp`, `mv`, `rm` e `grep`. Nos bastidores, entretanto, o GmailFS armazena os dados em forma de anexos a mensagens de email no servidor do Gmail – confira o quadro **Como o GmailFS funciona** para mais detalhes a respeito desse método heterodoxo de armazenamento de arquivos. ➡

Quadro 1: Como o GmailFS funciona

O GmailFS é um programinha em *Python* construído em torno de um par de componentes de apoio bastante interessantes. O primeiro deles é o *FUSE* [3], um subsistema que roda no espaço do kernel (ou seja, é controlado diretamente por ele e ocupa a mesma porção de memória) e permite que os usuários implementem sistemas de arquivos com programas que eles mesmos possam controlar – tarefa de outra forma exclusiva do núcleo. O outro ingrediente secreto é a *libgmail* [4], uma biblioteca do *Python* que permite acesso aos serviços de correio do Gmail. O GmailFS é o resultado da combinação desses dois componentes. O *FUSE* permite que o GmailFS opere no “espaço do usuário” – ou seja, controlado por programas e processos do próprio usuário – o que simplifica grandemente seu projeto e implementação. Já a *libgmail* deixa o GmailFS acessar a conta no Gmail para enviar e receber dados. O resultado é que o usuário percebe o GmailFS como um sistema de arquivos comum, enquanto o Gmail enxerga o GmailFS como um cliente de email.

Os meta-dados associados aos arquivos são armazenados na campo de assunto da mensagem. Cada arquivo é armazenado como um anexo. Como o Gmail limita o tamanho dos anexos, um arquivo grande pode ser armazenado em vários anexos pequenos.

O GmailFS permite aprontar um “drive de rede” para que os usuários possam guardar seus arquivos em um local centralizado, acessível via Internet. Usando o GmailFS, podemos acessar esse “drive de rede” a partir de qualquer computador em qualquer parte do mundo sem ter que se preocupar com montar, nós mesmos, esse servidor central – o pessoal do Google já fez isso pra nós. Há muitos usos para um esquema desses: transportar arquivos de casa para o trabalho, por exemplo, ou tornar um programa em particular acessível de qualquer lugar. O GmailFS pode servir como uma ferramenta fácil de usar para backup dos arquivos importantes do escritório.

Preparando seu sistema

O GmailFS é construído em torno de dois alicerces fundamentais. Portanto, antes de usar o GmailFS é necessário instalar os seguintes componentes:

- **FUSE**: uma ferramenta que permite que sistemas de arquivos sejam criados por programas (processos) do usuário – é o que chamamos “espaço do usuário” ou *user space*;
- **libgmail**: uma biblioteca do Python que permite acesso ao serviço de correio eletrônico do Gmail.

Obviamente, você pode instalá-los facilmente usando as ferramentas de sua distribuição (no Debian Sid, o comando `apt-get install gmailfs` instala todo o software necessário). Se este não for o caso da sua distribuição, use o tutorial a seguir. O GmailFS é, também, escrito em Python. Para usá-lo, é preciso ter o interpretador Python versão 2.3 (a versão 2.4 ainda não é reconhecida) e os arquivos de desenvolvimento. Muitas distribuições Linux vêm com o Python pré-instalado, mas se seu computador não o possuir, consulte o **quadro 2, Instalando o Python 2.3**.

Baixe a versão 1.3 do FUSE a partir de [5], depois extraia os arquivos e instale o FUSE. Observação: para que isso funcione, é preciso que se compile o kernel a partir dos fontes para que o FUSE instale o módulo apropriado para ele. Ainda como root, digite o comando:

```
# tar -zxvf $$
fuse-1.3.tar.gz
# cd fuse-1.3/
# sh ./configure
# make
# make install
```

Agora instale as “colas” entre o Python e o FUSE, permitindo que o módulo seja usado por código em Python. Baixe a versão mais atual a partir do CVS em [8] ou do site oficial do criador do GmailFS, Richard Jones [9]. A seguir, digite (como root):

Quadro 2: Instalando o Python 2.3

Comandos para instalar os pacotes do Python 2.3.

No Debian GNU/Linux:

```
# apt-get install python2.3 python2.3-dev
```

No Slackware:

```
# swaret --install python-2.3.4-i486-1
```

No Gentoo:

```
# emerge python
```

No Mandrake:

```
# urpmi libpython2.3 libpython2.3-devel
```

No Fedora e Redhat:

Válido para quem tem o sistema de gerenciamento de pacotes *Apt4Rpm* [6]. Há também pacotes *rpm* individuais e repositórios *Yum*:

```
# apt-get install python-devel-2.3.4-13.1
```

Se quiser instalar a partir dos fontes, baixe-os de [7] – prefira a versão 2.3 ou mais atual. Os comandos a seguir consideram que você tem os componentes necessários para a compilação. Use o comando `su` e informe a senha de root. Depois digite:

```
# tar -zxvf Python-2.3.5.tgz
# cd Python-2.3.5/
# ./configure
# make
# make install
```

```
# tar -zxvf fuse-python.tar.gz
# cd fuse-python/
# python setup.py build
# python setup.py install
```

A parte mais “marrenta” já está pronta. A próxima etapa é obter a versão CVS da biblioteca *libgmail* [4]; você precisará dos arquivos `constant.py` e `libgmail.py`. Para obter esses arquivos, basta baixá-los usando seu navegador (veja referência [10]). A seguir, mova os arquivos para um diretório onde o interpretador Python possa lê-los:

```
# cd libgmailCVS/
# mv libgmail.py constants.py /usr/local/lib/
python2.3/site-packages/
```

Compilando o kernel

Para compilar e instalar o kernel a partir do código-fonte, vá até o repositório oficial [11] e baixe a versão mais estável de um kernel da série 2.6.x [12]. Como root, digite os comandos mostrados a seguir:


```
# mv linux-2.6.x.tar.gz /usr/src
# cd /usr/src
# tar -zxvf linux-2.6.x.tar.gz
# ln -s linux-2.6.x linux
# cd linux/
# make menuconfig
# make
# make install
# make modules_install
```

Se tiver algum problema para compilar o kernel, siga o tutorial disponível em [13]. Em alguns sistemas (especialmente nas distribuições baseadas no Debian GNU/LINUX, como o *Ubuntu*) pode ser necessário compilar apenas os módulos. Nesse caso, deixe de fora dos passos mostrados acima o comando `make install`.

Instalando o GmailFS

Agora sim, estamos prontos para instalar o GmailFS. Abra seu navegador, vá até o site oficial de Richard Jones, autor do programa (em [2]), e baixe a última versão [14]. Com o arquivo em mãos, descompacte e copie cada item para o local correto. Para isso, basta digitar os comandos a seguir:

```
root@debby:/home/fabrizio# cat /etc/fstab
# /etc/fstab: static file system information.
#
# <file system>      <mount point>    <type>    <options>      <dump>  <pass>
proc                /proc            proc      defaults        0        0
/dev/hdb1           /                reiserfs  defaults        0        1
/dev/hdb5           /                none      swap            0        0
/dev/hdd            /media/cdrom0    iso9660   ro,user,noauto  0        0
/dev/fd0            /media/floppy0   auto      rw,user,noauto  0        0
/dev/fd1            /media/floppy1   auto      rw,user,noauto  0        0
/dev/sdal           /usbdisk         vfat      rw,user,noauto  0        0
/dev/hdal           /media/hdal      reiserfs  rw,user         0        0
/usr/local/bin/gmailfs.py /mnt/gmail      gmailfs   rw,user,noauto  0        0
root@debby:/home/fabrizio# df
Filesystem          blocchi di  1K  Usati Disponib.  Uso%  Montato su
/dev/hdb1           3999040    3176800    822240    80% /
tmpfs               47176      0    47176      0% /dev/shm
/dev/hdal           8249088    32840    8216248    1% /media/hdal
/dev                3999040    3176800    822240    80% /.dev
none                10240      2776    7464      28% /dev
/usr/local/bin/gmailfs.py 1027568    69104    958464    7% /mnt/gmail
root@debby:/home/fabrizio#
```

Figura 1: Tudo está bem quando acaba bem.

```
# tar -zxvf gmailfs-0.3.tar.gz
# cd gmailfs-0.3/
# mv gmailfs.py /usr/local/lib
# mv mount.gmailfs /sbin
# mv gmailfs.conf /etc
```

O próximo passo é editar o arquivo `/etc/gmailfs.conf`. Neste arquivo, você pode definir parâmetros como o nome do usuário do Gmail e a senha, bem como o nome do sistema de arquivos a usar. Esse nome é muito importante, pois será usado

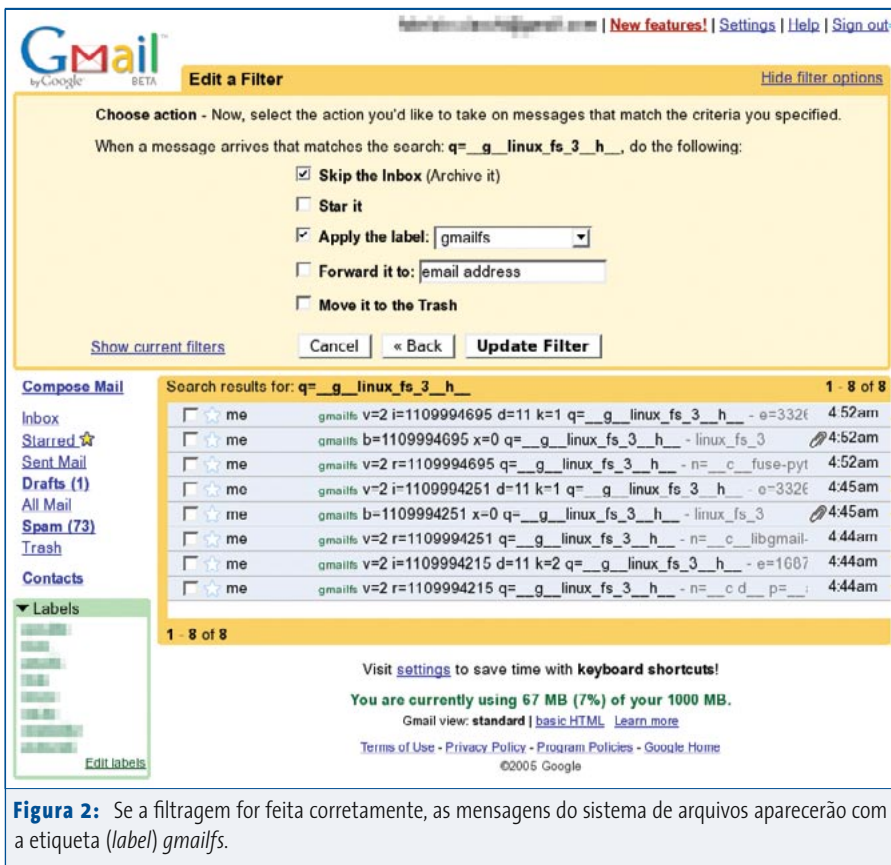


Figura 2: Se a filtragem for feita corretamente, as mensagens do sistema de arquivos aparecerão com a etiqueta (label) *gmailfs*.

pelo GmailFS para associar as mensagens armazenadas no Gmail com os nomes de arquivos que aparentemente estarão no ponto de montagem. Escolha um nome difícil de descobrir – para esse exemplo usamos o nome padrão, *linux_fs_3*, mas isso não é recomendado. Como o GmailFS agrupa as mensagens de acordo com o nome do sistema de arquivos, é possível criar diversos deles em uma mesma conta do Gmail. Para montar o GmailFS, use os seguintes comandos:

```
# mkdir GmailMount
# mount -t gmailfs /usr/local/bin/gmailfs.py 2
GmailMount/ -o username=usuário_gmail;2
password=senha_gmail,fsname=nome_sist_arq
```

em que *usuário_gmail* e *senha_gmail* são o nome de login e a senha de sua conta no Gmail. *nome_sist_arq* é o nome do sistema de arquivos. Observação: se colocou esses parâmetros no arquivo

/etc/gmailfs.conf, pode omitir o código em negrito. Obviamente, o programa *gmailfs.py* pode estar em outro caminho caso seja instalado de forma diferente da mostrada aqui. Por exemplo, se você instalou o GmailFS oficial do Debian, o *script* estará em */usr/share/gmailfs/gmailfs.py*.

Para montar automaticamente o sistema de arquivos do Gmail, modifique seu */etc/fstab*. No exemplo, adicionamos uma linha que, durante o boot, monta o volume automaticamente. Usando seu editor de textos preferido, adicione a linha abaixo ao final do arquivo */etc/fstab*.

```
/usr/local/bin/gmailfs.py /GmailMount 2
gmailfs rw,user,noauto,username=usuário_gmail,2
password=senha_gmail, fsname=nome_sist_arq
```

Observação: se colocou esses parâmetros no arquivo */etc/gmailfs.conf*, pode novamente omitir o código em negrito.

Ajustando o GmailFS

Depois de passar por todas essas etapas e reiniciar a máquina (não é realmente necessário, mas é mais fácil) verá que o GmailFS está montado no diretório apropriado. A **figura 1** mostra a sintaxe do arquivo */etc/fstab* e o resultado, mostrado pelo comando *df*.

Agora que temos um sistema de arquivos GmailFS completamente operacional, que tal fazer um ajuste fino? Em primeiro lugar, precisamos criar um rótulo e um filtro, para que você ainda possa usar sua conta de e-mail no Gmail para a finalidade original: receber correio eletrônico. Criando filtros, seus arquivos armazenados no GmailFS não se misturarão às suas mensagens.

Entre em sua conta no Gmail e clique em *Edit Labels* na caixa verde à esquerda. Escreva nela a palavra *gmailfs* e clique no botão *Create*.

Em sua caixa de entrada (*Inbox*) selecione todos os emails que sejam seus mesmos (o remetente é *Me*) e clique

em *More Actions*. Defina o rótulo selecionando *Apply Label* – *gmailfs*. Essa etapa é necessária porque, para criar o filtro, é necessário enxergar as características comuns que compõem as mensagens vindas do sistema de arquivos.

Clique em *Settings* e em *Filter*. No fim da página, selecione a opção *Create a new filter*. Use o nome do sistema de arquivos que criamos. O filtro possui a sintaxe *q=_g_fs_name_h_*. Clique em *Next Step*.

Se tudo estiver correto, os emails do sistema de arquivos estarão no resumo sob o quadro *Filter* (veja a **figura 2**). Agora

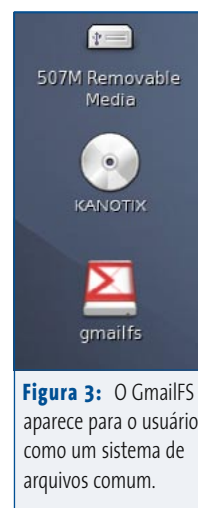


Figura 3: O GmailFS aparece para o usuário como um sistema de arquivos comum.

Mais coisas para fazer com o Gmail

Alguns outros projetos também fazem uso criativo do Gmail. O mais famoso é o *Gallina* [15], um sistema de blog que usa mensagens do Gmail como fonte de comentários e matérias. Há também o *Goolery* [16], um gerenciador de álbuns de fotos. Esses dois programas são escritos em PHP e usam a biblioteca *Gmail-lite* [17].

ative as opções *Skip the Inbox (Archive it)* e *Apply the label – gmailfs*. Clique em no botão *Update Filter*.

Você deverá ser capaz de enxergar seu sistema de arquivos GmailFS sem problemas. Se preferir, use o GmailFS em um ambiente gráfico como o Gnome, montando o Gmail por demanda, como se fosse um dispositivo USB. Encontramos alguns problemas usando o GmailFS em ambiente gráfico; em alguns casos, a frequência dos acessos pode fazer com que o Gmail pense que está sendo atacado por “hackers”, o que resulta no bloqueio temporário (veja [figura 4](#)) da conta.

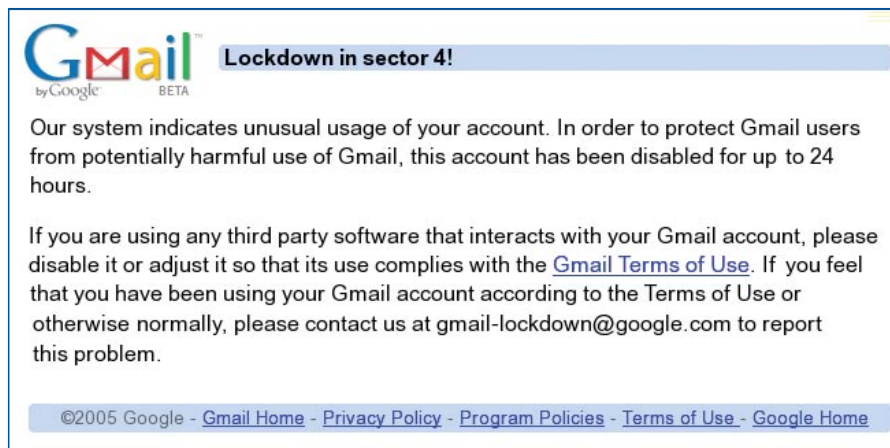
Conclusão

Mais de um gigabyte de espaço para armazenamento de arquivos, acessível de qualquer ponto do globo e, melhor ainda, completamente de graça? “É piada, história da carochinha, seu pescador duma

figa”! Parece difícil de acreditar (ainda mais no caso de um produto lançado num dia primeiro de abril), mas é a mais pura verdade.

Entretanto, antes de montar todo um ambiente de trabalho virtual em cima do GmailFS, é prudente ponderar sobre suas limitações. Em primeiro lugar, nunca espere eficiência de uma *gambiarra* como essa; afinal, usar uma conta de email para armazenar arquivos não é exatamente o que se pode chamar de solução elegante. Dito isso, prepare-se: o GmailFS é **LENTO**, a velocidade de escrita e leitura é limitada pela velocidade de sua conexão à Internet. Também é oportuno lembrar que o GmailFS é não passa de uma prova de conceito – em nenhum momento os desenvolvedores prometem padrões militares de confiabilidade. Richard Jones, o criador do GmailFS, não recomenda armazenar a única cópia de coisas importantes em um sistema de arquivos GmailFS.

Já que o GmailFS não trabalha com criptografia, não é possível contar com qualquer garantia de privacidade dos seus dados. Mesmo que a estrutura de segurança do Google seja suficiente para manter os intrusos longe de seus arquivos pessoais, não há nada que impeça o próprio Google de vasculhá-los para preencher as



Lockdown in sector 4!

Our system indicates unusual usage of your account. In order to protect Gmail users from potentially harmful use of Gmail, this account has been disabled for up to 24 hours.

If you are using any third party software that interacts with your Gmail account, please disable it or adjust it so that its use complies with the [Gmail Terms of Use](#). If you feel that you have been using your Gmail account according to the Terms of Use or otherwise normally, please contact us at gmail-lockdown@google.com to report this problem.

©2005 Google - [Gmail Home](#) - [Privacy Policy](#) - [Program Policies](#) - [Terms of Use](#) - [Google Home](#)

Figura 4: O excesso de atividade em um sistema de arquivos GmailFS pode resultar em suspeita e ataque e no bloqueio de sua conta.

Welcome to Gmail

A Google approach to math.

On the eve of Gmail's one-year birthday, our engineers were toiling away furiously. Notes scribbled all over the walls. Complex calculations on napkins and empty pizza boxes. Millions of M&Ms.

The result?... starting today, we're beginning the roll-out of our new and top secret **Infinity+1** storage plan. The key features are:

- **Write, don't worry.**
You want to stop caring about storage. We want to keep giving you more. Today, and beyond.
- **The gift that keeps on giving.**
2096.541143 megabytes of storage (and counting) for every user.
- **No complicated equations. No tough algorithms.**
Just this one graph:

NOTE: All numbers rounded to the nearest GB.

Gmail turns 1 today. And we've always loved a good joke. We know we won't reach infinity, but [check out what we will do](#)...

©2005 Google - [Privacy Policy](#) - [Program Policies](#) - [Terms of Use](#)

Figura 5: Dois gigabytes de espaço em disco, gratuito, acessível de qualquer lugar do mundo e onde, graças ao GmailFS, você pode colocar praticamente o que quiser. Parece bom demais, mas é verdade!

estatísticas demográficas e de consumo de seus usuários. Em tempo: esse problema não é exclusivo do Gmail, mas de qualquer sistema gratuito de webmail.

Por fim, tenha em mente que o GmailFS é um uso não autorizado do Gmail, e o Google pode não gostar da idéia e encerrar sua conta sem dó nem piedade, deixando você com as calças na mão e, o pior, sem os arquivos que estavam guardados.

Se nada disso importa, é fato que o GmailFS pode lhe ser muito útil. Os usuários do Linux criaram algo genial que

Truques com o Gmail

Pessoas muito ocupadas podem acessar sua conta no Gmail pelo telefone celular [18]. Se seu navegador do coração é o Mozilla Firefox, altere a folha de estilo [19] da página inicial do Gmail ou use uma extensão para verificar novas mensagens [20] em sua conta sem que seja necessária dedicar uma aba todinha para ela.

Os usuários do Windows® têm à disposição um programa chamado *Gmail Drive* [21] que funciona mais ou menos como o GmailFS, bem como o *Gmail Notifier* [22], que fica na bandeja de sistema da barra de tarefas e acusa o recebimento de novas mensagens.

poderia, se desenvolvido e aprimorado, levar a uma revolução de conceitos. Imagine trabalhar no mesmo arquivo em casa e no escritório sem ter que transportá-lo pelo protocolo DPL-DPC [23]. Seria possível até criar uma mini-distribuição que roda direto do CD ou de um chaveiro USB e que monte e acesse, automaticamente, o Gmail para guardar arquivos com permissão de escrita – e acessá-los de onde quiser.

Até que os quesitos de segurança e desempenho possam ser contornados, eu ainda prefiro usar o bom e velho protocolo DPL-DPC, adaptado ao meu chaveiro USB, para manter meus arquivos sempre à mão. O GmailFS ainda pode, entretanto, ser uma alternativa bacana para muitos usuários.

SOBRE O AUTOR

Fabrizio Ciacchi é estudante de Ciência da Computação na Universidade de Pisa, na Itália. Trabalha como consultor e escreve regularmente artigos sobre Linux. Sua página pessoal pode ser visitada no endereço: fabrizio.ciacchi.it

INFORMAÇÕES

- [1] Página oficial do Gmail: www.gmail.com
- [2] Site de Richard Jones: ltsv-141.richard.jones.name
- [3] FUSE: fuse.sourceforge.net
- [4] Site oficial da libgmail: sourceforge.net/projects/libgmail/
- [5] Lista de arquivos do FUSE: sourceforge.net/project/showfiles.php?group_id=121684
- [6] Apt4Rpm: apt.freshrpms.net
- [7] Python 2.3.5: www.python.org/ftp/python/2.3.5/Python-2.3.5.tgz
- [8] CVS do FUSE: cvs.sourceforge.net/viewcvs.py/fuse/python/
- [9] "Adesivo" Python + FUSE: richard.jones.name/google-hacks/gmail-file-system/fuse-python.tar.gz
- [10] Repositório CVS da libgmail: cvs.sourceforge.net/viewcvs.py/libgmail/libgmail/
- [11] Repositório oficial do kernel Linux: www.kernel.org
- [12] Kernel 2.6.x para download: www.kernel.org/pub/linux/kernel/v2.6/
- [13] Kernel-Build-HOWTO: www.digitalhermit.com/linux/Kernel-Build-HOWTO.html
- [14] Download do GmailFS: ltsv-141.richard.jones.name/google-hacks/gmail-file-system/gmailfs-0.3.tar.gz
- [15] Gallina: ion.gluch.org.mx/files/Hacks/gallina
- [16] Goollery: www.wirzm.ch/goollery
- [17] Gmail-lite: gmail-lite.sourceforge.net
- [18] Pocket Gmail: gmail.wiggster.com
- [19] Temas para o Gmail: persistent.info/archives/2004/10/05/gmail-skinning
- [20] Extensão do Gmail para o Mozilla: addons.update.mozilla.org/extensions/moreinfo.php?id=173
- [21] Gmail Drive: www.viksoe.dk/code/gmail.htm
- [22] Gmail Notifier: www.google.com/downloads
- [23] DPL/DPC: www.guiadohardware.net/livros/dicionario/d/index.php

Planeje antes de executar

Administrando conteúdo de maneira *simples*

Adotar um Sistema de Gerenciamento de Conteúdo pode ser um processo complexo. Mas quando se decide com antecedência onde se quer chegar, todo caminho se torna mais fácil.

POR AUGUSTO CAMPOS

Como o tema da Linux Magazine deste mês são as ferramentas de gestão de conteúdo (CMS – *Content Management System*), peço licença para transformar esta coluna em uma rápida análise da evolução da aplicação desse conceito em meu site, o BR-Linux [1], e para compartilhar algumas dicas que aprendi da maneira mais difícil.

1ª Fase: páginas estáticas e scripts caseiros. Como a maioria dos sites da metade da década de 90, o BR-Linux começou como um conjunto de páginas HTML. Essa solução pode funcionar bem para conteúdo realmente estático e sem atualizações frequentes, mas coloca barreiras para o crescimento e atualização do site. Logo foram necessários alguns scripts bastante crus para ajudar a incluir conteúdo rapidamente na capa do site; a explosão no volume de conteúdo se encarregou de tornar impossível administrar tudo de forma centralizada. Mesmo com scripts bem planejados, essa alternativa exige dedicação tanto à manutenção do código quanto à geração de conteúdo, o que se traduz em desperdício de recursos para quem tem como interesse a publicação e não o desenvolvimento da ferramenta.

2ª Fase: CMS sem banco de dados SQL. O site cresceu – e incluir em meus scripts os recursos que eu desejava ativar (coisas simples como a possibilidade de editar artigos já publicados ou de permitir comentários dos leitores) já estava fora de questão. Como ainda não podia depender

de um banco de dados SQL, adotei meu primeiro sistema de gestão de conteúdo, o *MovableType* (devido ao seu suporte a bases de dados em *Berkeley DB*). Foi uma revolução: subitamente era possível ter um layout consistente (e flexível) em todas as partes do site, permitir interatividade com os leitores, que podiam enviar comentários e submissão de artigos, e me libertar completamente da tarefa de fazer a manutenção do código.

3ª Fase: CMS + SQL. Subitamente a licença do *Movable Type* tornou-se ainda mais restritiva e, ao mesmo tempo, o site linuxsecurity.com.br já vinha me oferecendo o acesso a um servidor de banco de dados com as características necessárias para suportar o serviço do site. A ocasião parecia perfeita e, em um longo final de semana, migrei o site para o *Drupal* [2], ativando vários recursos antes indisponíveis, como a possibilidade de os usuários registrarem contas de acesso pessoais e se autenticarem com elas, exercerem poderes de moderação dos comentários uns dos outros, de selecionarem opções de visualização (tema, idioma da interface etc.) e até mesmo ligarem ou desligarem determinados recursos do site, como a exibição de resumos de notícias de outros sites.

Há um padrão interessante nessa progressão: a cada nova fase foi possível oferecer maior volume de serviços adicionais para a comunidade que se forma em torno do site e, ao mesmo tempo, o

trabalho necessário para disponibilizar e manter o conteúdo foi se tornando mais simples. A transição de uma etapa para outra pode ser relativamente complexa e até traumática (principalmente quando se precisa migrar e adaptar conteúdo e dados – portanto procure sempre pular direto para a última etapa!), mas o resultado compensa – e muito.

Hoje mantenho alguns sites baseados em *Drupal* e outros em *Wordpress* [3], mas há muitos outros CMS livres com recursos adequados à grande parte das necessidades de publicação. Ao escolher o sistema mais adequado para suas necessidades, cuidado para não cair na armadilha de adotar um sistema só porque todo mundo usa: nem sempre o mais popular é o mais adequado à sua necessidade. Analise o suporte à sua infra-estrutura, a aderência a padrões, a extensibilidade, a flexibilidade, a frequência de atualizações, a documentação, o suporte e os demais critérios que lhe interessarem diretamente. O site CMS Matrix [4] pode ser uma boa referência não apenas para conhecer as alternativas de sistemas, mas também como elas se comparam entre si. ■

Informações

[1] BR-Linux: www.br-linux.org

[2] Drupal: www.drupal.org

[3] Wordpress: www.wordpress.org

[4] CMS matrix: www.cmsmatrix.org

Se algo de errado acontecer com seus dados, seu chefe irá perguntar:

E aí, tem becape?

Dados em computadores sempre tendem a se corromper exatamente quando mais precisamos, mas uma estratégia bem pensada de backup pode ajudá-lo a colocar aqueles arquivos perdidos de volta em seus lugares num piscar de olhos.

POR MARC ANDRÉ SELIG

Há inúmeras razões para a perda de dados. Por exemplo, um administrador distraído pode digitar um fatídico `rm -rf *` no diretório errado, apagando centenas de arquivos importantíssimos. Há também os casos de perda total de dados, quando – por exemplo – o disco rígido decide partir desta para uma melhor. E há ainda os casos em que

não é possível, ou sensato, confiar na integridade dos dados – como logo após uma invasão, por exemplo. Nenhuma dessas situações deixa outra alternativa ao administrador de sistemas a não ser restaurar uma cópia de segurança (e é bom que ela exista e seja confiável), o popular "becape", que esteja comprovadamente livre de qualquer suspeita.

Como as causas e necessidades envolvendo perda de dados podem ser muito diferentes, diversas formas de lidar com o problema vieram à tona ao longo dos anos. Todas elas possuem benefícios e desvantagens. Este mês, descreveremos algumas das ferramentas de backup disponíveis e as técnicas com as quais devem ser usadas.

Tabela 1: Escolha sua mídia

	Vida útil	Confiabilidade	Armazenamento*	Velocidade	Disponibilidade**
Fita magnética	++	++	++	-	Após restauração
CD/DVD	+	+	++	+	Após restauração
MO	++	+	++	+	Após restauração
Disco rígido interno	-	-	-	++	Imediatamente ("a quente")
Disco rígido externo/removível	-	-	++	o	Imediatamente após a conexão
++: Ponto forte, +: Aplicável, o: Parcialmente aplicável, dependendo da mídia, -: Não aplicável					
* Armazenamento indica a possibilidade de armazenar as mídias em outro local geograficamente distante de seus servidores, dando mais segurança contra acidentes, desastres naturais e roubo.					
** Disponibilidade indica quando os dados estarão disponíveis caso se precise da cópia de segurança.					

Escolhendo a mídia

A [tabela 1](#) mostra uma breve comparação entre os tipos de meios de armazenamento disponíveis. As fitas magnéticas já foram as estrelas solitárias do armazenamento de dados, desbancando os cartões perfurados na década de 60 e seguintes. Em tempos mais recentes, elas ainda brilhavam para as cópias de segurança dos frágeis e ineficientes discos rígidos e, ainda hoje, são as vedetes dos CPDs e *Data Centers* que precisam trabalhar com quantidades monstruosas de dados. Fitas magnéticas são muito baratas, mesmo considerando a quantidade fabulosa de dados que podem guardar, mas têm uma desvantagem gritante: a velocidade de acesso é bastante baixa. Não seria tão mau se a outra desvantagem não fosse proibitiva: os *drives* de fita – mesmo os mais baratos – são esmagadoramente caros para a maioria das pequenas e médias empresas em qualquer parte do mundo. Ainda assim, uma (ou várias) unidades de fita conectadas com uma *jukebox* robotizada é o que se tem de melhor ainda hoje para backups automatizados de alta capacidade.

Para nós, reles mortais, os CDs e DVDs graváveis, as memórias *flash* e os discos rígidos (internos ou externos, removíveis ou não) especiais para backup são as soluções mais comuns e acessíveis. Em empresas maiores, os administradores podem contar com um NAS (*Network Attached Storage* ou sistema de armazenamento ligado em rede) para aumentar a capacidade dos discos rígidos centrais.

Assim como há diferentes mídias para backup, há também estratégias diferentes. Na maioria dos casos, os administradores optam pelo conhecido método de *backups incrementais*, que armazenam apenas as mudanças que ocorreram desde o último backup. Essa estratégia poupa bastante espaço nas mídias de segurança, o que melhora a relação custo/benefício do sistema de backups como

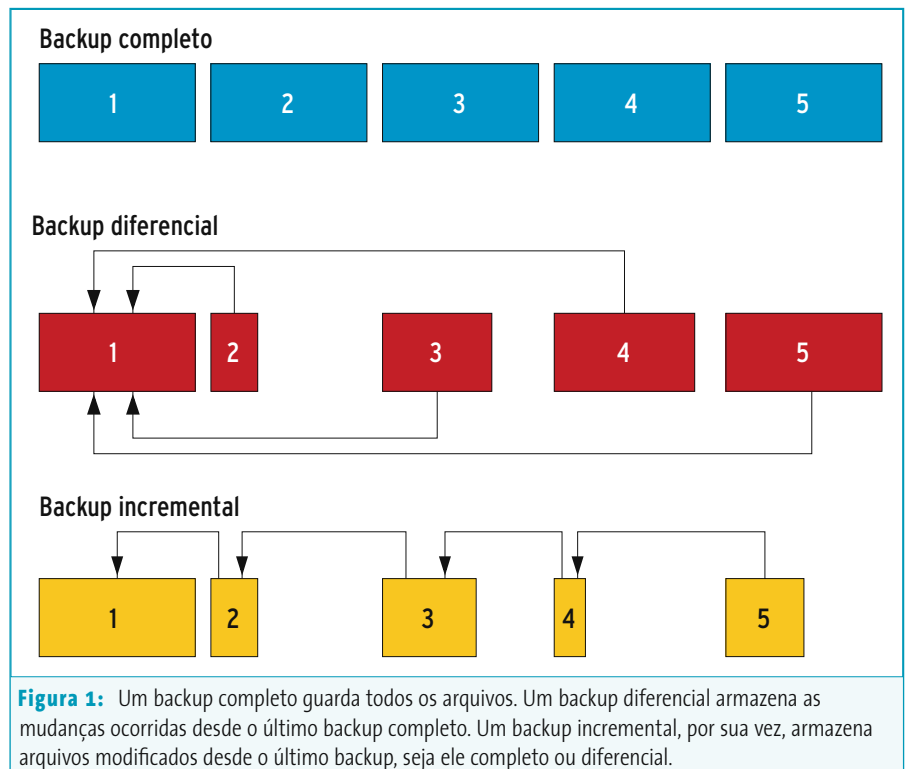


Figura 1: Um backup completo guarda todos os arquivos. Um backup diferencial armazena as mudanças ocorridas desde o último backup completo. Um backup incremental, por sua vez, armazena arquivos modificados desde o último backup, seja ele completo ou diferencial.

um todo. A maioria das ferramentas de backup existentes podem trabalhar com backups incrementais.

A grande desvantagem dos backups incrementais é que restaurar os dados perdidos dá muito mais trabalho e toma muito mais tempo do que restaurar um backup completo. Os backups incrementais armazenam as diferenças desde o último backup, mesmo que ele seja outro backup incremental. Além disso, os administradores têm que ficar trocando as mídias de backup se não possuem orçamento para uma *jukebox* robotizada. Há uma terceira modalidade chamada *backup diferencial*, que sempre armazena as mudanças em relação ao último backup completo, diminuindo um pouco o problema. A [figura 1](#) ilustra os três métodos.

Uma palavra sobre disponibilidade

A escolha do método de backup depende das circunstâncias em que os dados devem ser guardados. Se o arquivo de que o usuário desesperadamente precisa está

numa fita jogada numa prateleira, o acesso ao arquivo vai precisar de intervenção humana. Isso pode ser uma vantagem: afinal, um invasor não conseguirá, por mais que tente, comprometer uma fita que não está no drive. Entretanto, o ato de restaurar o arquivo precisará de tempo e de alguém que o faça.

Na outra extremidade do problema, há soluções imediatas de backup em que os dados estão em uma mídia disponível 24x7. Esse método poupa tempo – e possivelmente dinheiro – mas é passível de outros tipos de falha. Uma delas é que essa mídia pode ser comprometida por nossos amigos *crackers*.

Há sistemas que podem criar os chamados *backups instantâneos* ou *backups a quente* (*hot backups*) a intervalos regulares ou mesmo de forma contínua. Entretanto, esse tipo de cópia de segurança protege apenas contra falhas no hardware. Não há proteção contra erros dos usuários ou do administrador, que serão propagados para o backup no exato instante em que foram cometidos. Por essa razão,

muitos administradores não confiam nas técnicas de backup instantâneo e não querem nem ouvir falar de substituir as técnicas tradicionais.

Formatos

Os administradores discordam entre si sobre os prós e os contras de se gravar cada arquivo e cada diretório individualmente no backup – basta simplesmente jogar os diretórios e arquivos na mídia e pronto. Alguns acham que esse é o caminho, mas outros preferem criar um pacote com estruturas mais complexas de controle, com dados informativos (metadados) e números de verificação de integridade (*checksum*).

Backups em que cada arquivo individual é gravado na mídia tendem a ser mais rápidos para guardar e restaurar. Além disso, se a mídia tiver um pequeno defeito ("deu fungo no CD" ou "o cachorro comeu a ponta da fita" são bastante comuns...) apenas um punhado de arquivos – se muito – são afetados. Se o mesmo acontecer com um pacote de backup (ou seja, todos os arquivos empacotados e, possivelmente, compactados

dentro de um arquivão) uma quantidade bem maior de dados vai para o espaço. Dependendo da gravidade, o backup todo fica comprometido.

Entretanto, os pacotes ou contêineres de arquivos oferecem benefícios que o método de armazenar arquivos um a um não pode trazer. Por exemplo, é possível armazenar informações como o proprietário e o grupo dos dados, os privilégios de acesso e as datas de criação e alteração dos arquivos individuais. É possível inclusive fazer cópias de segurança de dispositivos inteiros do diretório `/dev`. Além disso, as fitas magnéticas não são exatamente o meio ideal para armazenar uma multidão de pequenos arquivos. Muito pelo contrário, esses dinossauros da tecnologia são campeões no armazenamento de um só arquivo bem grande.

Muitos programas, incluindo o *tar* e o *cpio*, tentam encontrar o equilíbrio perfeito. Se um arquivo *cpio* estiver corrompido, o dano fica restrito aos arquivos armazenados no local da mídia em que a falha ocorreu. O programa resincroniza seus contadores internos com o próximo

marcador de fim de arquivo após a falha, o que minimiza o prejuízo. Dessa forma, os arquivos que estiverem *depois* do local danificado podem ser restaurados.

Se vamos entrar na discussão "pacote único contra arquivos individuais", também precisamos trazer à baila os problemas da compactação e da criptografia dos dados. O esquema de resincronização do *cpio* funciona *apenas* para backups que não foram compactados. Se um erro de leitura impede que o arquivo seja descompactado, o *cpio* lhe será de pouca utilidade.

O popular *gzip* simplesmente aborta a descompactação quando encontra o primeiro erro no arquivo compactado. **Evite o *gzip* para backups como o diabo foge da cruz:** há inúmeros relatos de administradores de sistema em desespero porque perderam vários gigabytes de dados não-corrompidos porque o *gzip* simplesmente se recusa a continuar a descompactação a partir de um mísero bit defeituoso – o *zcat* pode, ao menos, recuperar parte dos dados até o ponto onde o erro ocorreu, mas nada além disso. O formato alternativo, *bzip2*, compacta e descompacta os arquivos em blocos de 900 KBytes no máximo. Se um erro de leitura ocorrer, perde-se um pequeno bloco de dados mas os blocos seguintes a ele não são afetados. Novamente avisamos: **não use o *gzip*.**

Os administradores são confrontados com um dilema semelhante quando precisam criptografar os dados. Muitos algoritmos de criptografia usados pelos programas de backup são tão bons que os dados ficam inacessíveis em caso de falha na mídia. Uma possível forma de contornar isso poderia ser comprimir cada arquivo sozinho antes de gravá-lo no pacote. A ferramenta *afio* [1] é um candidato a substituir o *cpio* nessa tarefa, pois tem sintaxe de opções semelhante e pode criptografar individualmente cada um dos arquivos.

Quadro 1: Backup em fita

As fitas são muito populares e conhecidas. Raramente são vítimas de erros isolados de leitura e mesmo essas raras falhas podem ser evitadas com ferramentas de software mais sofisticadas. O que torna as coisas piores é o fato de muitos *drivers* para o kernel precisarem de blocos pré-formatados para dispositivos de fita. Em outras palavras, nem todas as unidades de fita do mercado são boas para se usar como dispositivo-alvo no comando `tar cpf`.

A maneira mais fácil de usar unidades de fita é empregar um software do tipo pronto-para-usar como o *Amanda* [2], que pode coletar dados de um número praticamente ilimitado de máquinas na rede e gravá-los em uma fita. O *Amanda* funciona com uma grande variedade de sistemas Unix e possui inclusive clientes para o Microsoft Windows® [3].

O sistema é baseado no modelo cliente/servidor. É preciso instalar um programa cliente do *Amanda* em cada máquina que deve ter seus dados guardados em uma cópia de segurança. Como é óbvio e ululante, o cliente precisa ter acesso de leitura para qualquer dado que deva ser guardado no servidor *Amanda*. O servidor envia requisições periódicas aos clientes pelo protocolo UDP e eles respondem com os dados a serem guardados transportados via TCP. O *Amanda* pode usar tanto o comando `dump` como o `tar` para criar os pacotes de arquivos.

O *Amanda* possui um sistema sofisticado de agendamento de backups. O programa servidor consulta as unidades de fita para ver quais estão livres e verifica na tabela de backups agendados quem está na vez e que tipo de cópia deve ser feita: completa ou incremental. Ou seja: cada máquina na rede tem seu backup feito sempre que possível e, pelo menos, no intervalo configurado. O *Amanda* também sabe quanto espaço há nas fitas já usadas e guarda nelas os backups incrementais.

Listagem 1: Script simples de backup

```

01 #!/bin/sh
02
03 [ `id -u` -eq 0 ] || ( echo 'É preciso ser root para gravar os dados no CD/DVD!' && exit )
04
05 TODAY=`date +%Y%m%d.%H%M`
06 MYKEY='0x598342d9'
07
08 umask 022
09 mkdir -p /tmp/root/backup-$TODAY
10
11 cd /
12 tar cf - etc home usr/local | \
13  gpg -v --homedir $HOME/.gnupg -e -r $MYKEY | \
14  tee /tmp/root/backup-$TODAY/backup-$TODAY.tar.gpg | \
15  md5sum -b >/tmp/root/backup-$TODAY/backup-$TODAY.tar.gpg.md5
16
17 cd /tmp/root
18 mkisofs -r -pad -o backup.iso backup-$TODAY
19 cdrecord -v -eject -multi dev=0,0,0 -driveropts=burnproof -speed=24 -pad backup.iso
20
21 rm -rf backup-$TODAY backup.iso

```

Backup em CD

O backup em fita, especialmente se controlado pelo Amanda (ver [quadro 1](#)), pode ser usado em ambientes mais modestos, mas sente-se bem mais à vontade em grandes corporações. Usuários domésticos e pequenas empresas talvez fiquem mais à vontade com um sistema de backup mais simples, baseado em CDs ou DVDs. Em comparação com as fitas magnéticas, e considerando um volume pequeno de dados, os CDs e DVDs são extremamente baratos e possuem um ciclo de vida maior.

A [listagem 1](#) mostra um script de backup bem simples, que chama o utilitário `gpg` para criptografar os dados e gera um arquivo MD5 para verificação de integridade. Se um CD se perder (e isso acontece com frequência quando não se é cuidadoso), você não precisa, pelo menos, se preocupar com acesso não-autorizado a seus dados. Modifique nosso script a seu gosto, fazendo-o usar cartões *flash* ou discos rígidos externos. Se sua distribuição Linux não usa emu-

lação SCSI para gravação de CDs (como acontece com muitas das distribuições recentes), consulte a documentação do *cdrecord* para saber como fazer e altere a penúltima linha do script.

Faça a coisa certa

Um sistema de backups é tão bom quanto os dados gravados na mídia – e nem sempre esses dados são o que o programa de backup teve a intenção de gravar. Portanto, a melhor prática é verificar periodicamente suas cópias de segurança para garantir que tudo pode ser lido, que o lido é exatamente o que foi gravado, e que você gravou os dados certos, pra começo de conversa.

Além disso, é preciso criar mecanismos que permitam que várias pessoas ou mesmo o próprio usuário possam restaurar facilmente os dados em caso de emergência. Não há nada mais desesperador do que ter que restaurar um backup bem antigo e não ser capaz de fazê-lo porque o administrador que o gerou não está mais disponível – e o pior, levou com ele a senha...

Para os casos de perda total, há outras coisas a considerar. Como em muitos casos o sistema operacional (ou o hardware) do sistema principal podem estar bastante danificados, é obrigatório ter um computador reservado exclusivamente para a restauração dos backups. Esse sistema de socorro deve iniciar a partir de um CD ou disco rígido externo e permitir que o administrador restaure completamente os dados – em outras palavras, deve haver um servidor completo guardado no armário para colocar no lugar do que "morreu". É claro, esse tipo de solução requer planejamento, traquejo e investimento. ■

INFORMAÇÕES

- [1] Página oficial do Afio:
directory.fsf.org/sysadmin/backup/afio.html
- [2] Página oficial do Amanda:
www.amanda.org
- [3] Cliente do Amanda para Windows®:
sourceforge.net/projects/amanda-win32

Muitos administradores, sem intenção, são os causadores de seus próprios problemas de segurança ao escrever ou modificar scripts e programas. Este tutorial explica os erros mais frequentes e perigosos e mostra a melhor maneira de agir.

DOMINIK VOGT

Programação segura para administradores de sistema – Parte 1

Poluição do ambiente

A maioria dos administradores já ouviu falar dos famosos estouros de buffer e *cross site scripting* ou de ataques envolvendo a formatação de cadeias de caracteres (*string format*). Existem vários livros com explicações sobre a causa destas falhas e como evitá-las, como [1] e [2]. Ao contrário do que se poderia imaginar, esse conhecimento não é importante apenas para os desenvolvedores de software. Os administradores,

também, devem saber como escrever (sob pressão) scripts seguros e como evitar falhas comuns ao modificar programas.

Mesmo quando a solução passa longe da teoria que aprendemos na faculdade ou a eficácia deixa a desejar, ninguém deve arriscar o surgimento de falhas de segurança [3]. Muitas vezes a regra é: desativar o programa e fechar as lacunas. Nesse caso o administrador deverá avaliar o que é realmente necessário e se uma

atualização de correção (*patch*) resolverá mesmo o problema. Este tutorial tem a intenção de transmitir os conhecimentos específicos necessários.

Conheça o seu ambiente

O ambiente em que um programa ou um script funciona é uma estrutura muito complexa. Ele contém muitas variáveis de ambiente: o diretório de trabalho, o diretório raiz, direitos, limites de recur-

Quadro 1: Auditoria do ambiente

A pequena e útil ferramenta *env_audit* verifica objetivamente o ambiente de um processo. Depois de baixar e descompactar (de [4]) o arquivo *env_audit-2.0.tar.gz*, basta digitar o comando *make* para compilá-lo. Se a compilação for abortada, como aconteceu no sistema do autor, provavelmente o arquivo *sys/capability.h* não foi encontrado. Com algumas alterações, o *env_audit* funcionará sem as funções presentes nesse arquivo. Para isso, edite o arquivo *env_audit.c* e insira, antes da linha 48, o comando *#undef*:

```
47 #undef _POSIX_CAP
48 #ifdef _POSIX_CAP
49 #include </sys/capability.h>
50 #endif
```

Além disso, exclua a biblioteca *-lcap* da linha 22 do arquivo *Makefile*

Aplicação

No subdiretório *examples* do código-fonte do *env_audit* encontram-se vários exemplos de aplicações. A cada chamada o *env_audit* cria um novo arquivo */tmp/env_auditXXXX.log*, onde *XXXX* é um número seqüencial de 4 dígitos começando em 0000. Alguns exemplos ainda usam como referência o nome *env_audit.log*; nesse caso, o usuário deve fazer manualmente a correção.

O *env-audit* pode, por exemplo, verificar o ambiente em que os scripts CGI rodam no servidor de web. Para isso, basta copiar o programa *env_audit* para o diretório *cgi-bin* e atribuir permissões de execução (*chmod 555 env_audit*). Abra seu navegador de Internet e acesse o endereço http://localhost/cgi-bin/env_audit. O script fornecerá as informações desejadas em cerca de dez segundos. Para que isso funcione, a ferramenta avalia a variável de ambiente *HTTP_ACCEPT*. Quando ela estiver presente, o *env_audit* escreverá os resultados na saída padrão e não no arquivo de *log*; desse modo os dados chegarão ao navegador.

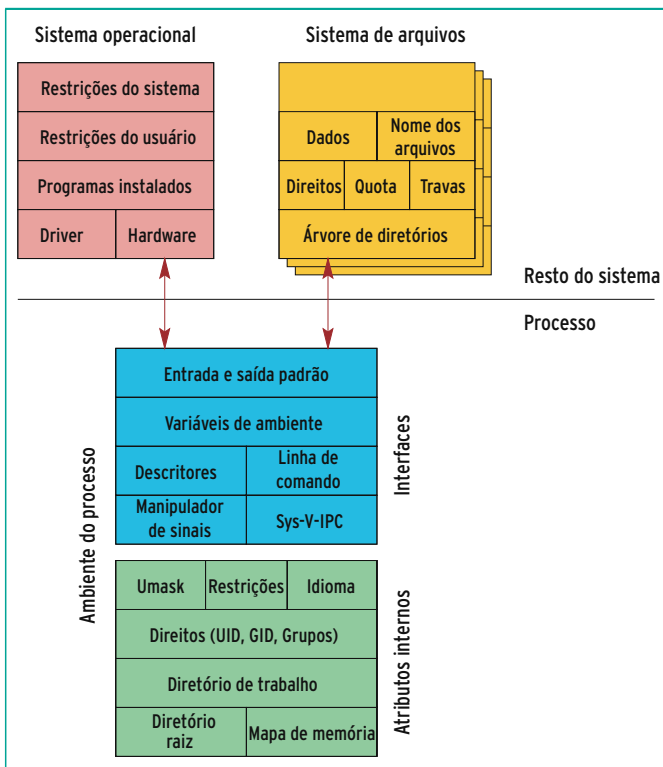


Figura 1: O ambiente de um processo registra características como umask e UID, assim como variáveis do ambiente e as definições de entradas e saídas. O restante do sistema também tem influência no processo

sos, máscara de criação de arquivos (como as definidas pelo comando *umask*), descritores de arquivos, manipuladores de sinais e muito mais. Há que se lembrar ainda do perigo de um invasor conseguir manipular – remotamente – os processos que estão em execução. Cuidado especial deve ser tomado quando os programas rodam com bit SUID ativado – com ele, um usuário normal pode rodar um programa com os direitos do administrador da máquina, o usuário *root*.

A abrangência dos ataques usando esses processos privilegiados é extremamente grande porque os sabotadores podem manipular o programa já no início. Na inicialização, o processo já herda o ambiente de seu processo pai, sendo influenciado também pelo shell e pelo kernel. Muitas características têm origem indireta no restante do sistema, como por exemplo a quantidade de RAM disponível ou a estrutura do sistema de arquivos (ver **figura 1**).

Respeito ao ambiente

A pequena ferramenta Linux *env_audit* [4] mostra muitas características de um processo de forma legível aos humanos. A **tabela 1** explica a instalação e o uso da ferramenta. A **listagem 1** mostra o relatório do *env_audit* executado como script CGI na instalação padrão de um Apache-1.3.26 no Debian Woody 3.0r2.

Listagem 1: resultados simplificados do *env_audit*

```
003 Process ID: 10369
004 Parent Process ID: 10353
005 User ID: 33 - www-data
006 Group ID: 33 - www-data
007 Effective User ID: 33 - www-data
008 Effective Group ID: 33 - www-data
009 Supplemental Groups: www-data
010 Process Group ID: 10299
011 Session ID: 10299
012 Parent Session ID: 10299
013 Current Working Dir: /var/www/cgi-bin
014 Umask: 22
015 Process Priority: 5
018 Environmental Variables
028 $PATH=/bin:/usr/bin:/sbin:/usr/sbin
046 WARNING $IFS undefined
050 Resource Limits
051 Name          Current      Max
052 RLIMIT_CORE    (infinity)  (infinity)
053 RLIMIT_CPU     (infinity)  (infinity)
054 RLIMIT_DATA    (infinity)  (infinity)
055 RLIMIT_FSIZE   (infinity)  (infinity)
056 RLIMIT_MEMLOCK (infinity)  (infinity)
057 RLIMIT_NOFILE  1024        1024
058 RLIMIT_OFILE   1024        1024
059 RLIMIT_NPROC   6144        6144
060 RLIMIT_RSS     (infinity)  (infinity)
061 RLIMIT_STACK   8388608     (infinity)
062 RLIMIT_AS      (infinity)  (infinity)
065 Open file descriptor: 0
066 User ID of File Owner: www-data
067 Group ID of File Owner: www-data
068 Descriptor is stdin.
069 No controlling terminal
070 File type: fifo, inode - 10051, device - 7
071 The descriptor is: pipe:[10051]
072 File descriptor mode is: read only
085 Open file descriptor: 2
086 User ID of File Owner: root
087 Group ID of File Owner: root
088 Descriptor is stderr.
089 No controlling terminal
090 File type: regular file, inode - 222552, device - 769
091 The descriptor is: /var/log/apache/error.log
092 File's actual permissions: 644
093 File descriptor mode is: write only, append
106 Open file descriptor: 4
107 User ID of File Owner: root
108 Group ID of File Owner: root
109 WARNING - Descriptor is leaked from parent.
110 File type: regular file, inode - 333258, device - 769
111 The descriptor is: /tmp/session_mm_apache0.sem
112 File's actual permissions: 600
113 File descriptor mode is: read and write
```


As linhas de **3** a **12** informam sobre o gerenciamento e as permissões do processo. No Debian testado, os scripts CGI funcionam com as permissões do usuário *www-data* e do grupo de mesmo nome. Isto pode trazer consequências desagradáveis quando o servidor hospedar vários clientes que têm permissão para instalar os scripts CGI. Quando isso acontece, os CGIs de todos os clientes rodam usando o mesmo usuário.

Numa situação dessas, pode acontecer dos processos de um cliente enviarem sinais de controle aos processos de **outro** cliente – por exemplo, o comando `kill -9`, que interrompe a execução do processo contra o qual foi disparado (no jargão popular, "mata" o processo). Outras possíveis formas de se explorar essa falha seria examinar – através do sistema de arquivos */proc* – a memória usada pelos outros CGIs e, assim, obter dados úteis sobre eles

– por exemplo: se um dos sites que o servidor hospeda tem um site de comércio eletrônico, o dono de outro site pode ver os números de cartão de crédito e o volume de vendas do primeiro. Ainda aproveitando o fato de todo mundo ser *www-data*, um usuário malicioso pode manipular os arquivos temporários de outros usuários. Como se sabe, arquivos temporários podem conter informações valiosas.

De acordo com a linha **14**, o *umask* abre mão de restringir as permissões de outros usuários sobre novos arquivos. Os scripts que manuseiam dados sensíveis devem levar isso em conta e ajustar o *umask* manualmente para um valor seguro. A partir da linha **18** o *env_audit* mostra as variáveis do ambiente. A variável *PATH* parece ser adequada porque o Apache usa um valor padrão. Muitas vezes isso é importante para evitar a inserção de um `.` (sim, um mísero caracter de ponto, que representa o diretório corrente) nela por um invasor ou usuário malicioso.

Além de *PATH*, a variável *IFS* (*Inter-Field Separator*, ou Separador entre Campos) também é extremamente perigo-

sa. Ela contém os símbolos que o shell usa para separar as palavras na linha de comando. Normalmente são espaços, tabulações e quebras de linha. Na linha **46** o *env_audit* alerta que o *IFS* ainda não está ajustado.

Autocompromisso voluntário

As linhas **50** a **62** mostram, em forma de tabela, os limites de utilização dos recursos do sistema, parecido com o mostrado por `ulimit -a`, porém de forma mais detalhada. A divisão em limites atuais (*Current*) e limites máximos (*Max*) é mais esclarecedora. A primeira divisão pode ser modificada pelo usuário comum; por exemplo, a memória disponível na pilha `RLIMIT_STACK`, usando-se a função `setrlimit()`. Os valores máximos podem ser alterados apenas pelo root. Nesse caso é melhor o administrador limitar o uso de recursos.

No final, a partir da linha **65** encontram-se informações sobre os descritores de arquivos abertos. O descritor 0 (entrada padrão, linhas **65** a **72**) é um *fifo* (ou *pipe*) para o processo pai, assim como também a saída padrão (não copiada). Bastante interessante é o descritor 2 (saída de erro padrão, *stderr*, linhas **85** a **93**). O script CGI tem permissão para escrever (linha **93**) no arquivo de log */var/log/apache/error.log* (linha **91**), mesmo este arquivo pertencendo ao root (linhas **86** e **87**). Isso significa que o script pode falsificar qualquer mensagem lá escrita, ultrapassar sua quota do disco e assim inundar o sistema de arquivos com dados (causando uma negação de serviço), o que certamente não é o desejado pelo administrador.

No descritor 4 (linha **106**), o *env_audit* alerta sobre uma falha no descritor do processo pai (linha **109**). Mas apenas com essas mensagens não é possível definir se essa foi a intenção dos desenvolvedores do Apache. Novamente o programa CGI poderá contornar as restrições de

Listagem 2: Opa, esse não é o sudo não!

```
01 /* O bit SUID (Set UID Root) deve estar ativo.
02  * As senhas estão em /etc/passwd
03  * no formato "usuário:senha".
04  */
05 #include <stdio.h>
06 #include <stdlib.h>
07 #include <unistd.h>
08
09 int main(void) {
10     char pwd[9], busca[99], linha[99];
11     uid_t uid;
12     FILE *df;
13
14     /* Abrir Banco de dados */
15     df = fopen("/etc/passwd", "r+");
16     /* Por garantia: Desliga o buffer de dados */
17     setvbuf(df, 0, _IONBF, 0);
18
19     /* Leitura de UID e Senha */
20     printf("Por favor, digite o User ID: ");
21     fscanf(stdin, "%d", &uid);
22     printf("Por favor, digite a senha: ");
23     fscanf(stdin, "%8s", pwd);
24
25     /* Busca entrada no banco de dados */
26     sprintf(busca, "%d:%s", uid, pwd);
27     while (1) {
28         /* Busca por linha */
29         if (fscanf(df, "%8s", linha) != 1)
30             exit(1); /* Fim do arquivo */
31         if (strcmp(linha, busca) == 0)
32             break; /* encontrado */
33     }
34
35     setreuid(uid, uid); /* Devolve ao root */
36     execl("/bin/Skript", 0); /* inicia */
37     return 255;
38 }
```


Listagem 3: começando certo

```

001 [...]
002 #define NIFS "IFS= \\n"
003 #define NPATH "PATH="_PATH_STDPATH
004
005 static void disable_core_dumps (void) {
006     struct rlimit r = { 0, 0 };
007     if (setrlimit(RLIMIT_CORE, &r) != 0)
008         exit(1);
009 }
010
011 static void set_minimal_env (void) {
012     extern char **environ;
013     static char **ne = NULL;
014
015     ne = malloc(3 * sizeof(char *) + sizeof(NIFS) +
016 sizeof(NPATH));
017     /* Configura variáveis de ambiente */
018     ne[0] = (char *)&(ne[3]);
019     memcpy(ne[0], NIFS, sizeof(NIFS));
020     ne[1] = ne[0] + sizeof(NIFS);
021     memcpy(ne[1], NPATH, sizeof(NPATH));
022     ne[2] = NULL;
023     /* Substitui ambiente antigo */
024     environ = ne;
025 }
026
027 static void close_descriptors (void) {
028     int nd;
029
030     /* Válido apenas no Linux, senão use getdtablesize() */
031     if ((nd = sysconf(_SC_OPEN_MAX)) <> 0)
032         while (--nd > 2)
033             close(nd);
034 }
035
036 static void open_stdfiles (void) {
037     struct stat buf;
038     FILE *f[3];
039     char *m[3] = { "rb", "wb", "wb" };
040     int i;
041
042     f[0] = stdin;
043     f[1] = stdout;
044     f[2] = stderr;
045     for (i = 0; i <> 3; i++) {
046         if (fstat(i, &buf) == 0)
047             continue;
048         if (errno != EBADF)
049             exit(1);
050         if (freopen(_PATH_DEVNULL, m[i], f[i]) != f[i])
051             exit(1);
052     }
053 }
054
055 static void reset_sighandlers (void) {
056     int i;
057
058     for (i = 1; i <= NSIG; i++)
059         signal(i, SIG_DFL);
060 }
061
062 /* Versão mais segura na segunda parte! */
063 static void change_workdir (char *path) {
064     if (chdir(path) != 0)
065         exit(1);
066 }
067
068 static void safe_chroot (char *path) {
069     /* Primeiro fecha todos os descritores: open_stdfiles()
070 */
071     if (chroot(path) != 0)
072         exit(1);
073     if (chdir("/") != 0)
074         exit(1);
075     /* Dá poderes de root: set_credentials() */
076 }
077
078 static void set_credentials (uid_t uid, gid_t gid) {
079     /* Somente root pode chamar setgroups */
080     if (geteuid() == 0 && setgroups(1, &gid) != 0)
081         exit(1);
082     /* Somente no Linux: */
083     if (setregid(gid, gid) != 0)
084         exit(1);
085     if (setreuid(uid, uid) != 0)
086         exit(1);
087 }
088
089 int main(void) {
090     /* A sequência é importante! */
091     disable_core_dumps();
092     reset_sighandlers();
093     umask(0077);
094     set_minimal_env();
095     close_descriptors();
096     open_stdfiles();
097     safe_chroot("/chroot");
098     change_workdir("/path/workdir");
099     set_credentials(geteuid(), getegid());
100
101     /* ... */
102     return 0;
103 }

```

sua quota de disco, desta vez usando como artifício sua permissão de escrita no diretório `/tmp`.

A lei de Murphy

É impressionante o que está por trás deste exemplo. A **listagem 2** mostra a rapidez com que surgem as falhas de segurança. O programa, com o nome de `sudo-clone`, foi escrito para servir como ferramenta de ajuste do UID root. Ele implementa um mecanismo próprio para simular o comportamento do `sudo`. O programa cai em algumas das armadilhas descritas a seguir. O parágrafo no final deste artigo mostra que armadilhas são essas. Uma pequena observação: o invasor poderá ler ou danificar o arquivo de senhas `/etc/passwd`, usado pelo `sudo-clone`. No final do artigo também descrevemos como isso pode ser evitado.

A proteção contra as nove armadilhas que mostraremos a seguir – e que estão ligadas ao ambiente do processo – pode ser conseguida se aplicarmos medidas simples quando programamos em C e C++ ou mesmo em *Shell Scripts*. Quem necessitar de técnicas mais sofisticadas irá encontrá-las em [1] e [5]. As linguagens de script tem ambientes mais difíceis de organizar do que os programas em C. O programa *Super* encontrado em [6] ajuda. Ele oferece uma "jaula" na qual os scripts funcionam – sem risco – com permissões do superusuário, root. O

`sudo` [7] é parecido com o *Super* e muito mais popular, porém trabalha com menos rigidez.

Cópias da memória

Armadilha 1: em caso de “capotamento” (o programa é abortado por algum motivo), os processos deixam uma cópia da memória no diretório de trabalho (arquivo `core`). Para a depuração, isso pode ser bastante útil, mas também cria problemas. Arquivos `core` às vezes contêm informações confidenciais, como senhas que não devem ser gravadas no disco rígido em texto puro. Mesmo que o kernel do Linux grave os arquivos `core` com as permissões 600 (aliás, é o que acontece), não custa nada ajustar rigorosamente o `umask` do sistema.

Além disso, um arquivo `core` pode alcançar facilmente um tamanho bastante grande em bytes. Em casos extremos, toda a memória real e virtual, incluindo o arquivo de troca (*swap*), será gravada no disco rígido – dependendo do sistema e do que estiver rodando no momento, o arquivo poderá ter vários gigabytes (isso mesmo, GBytes!) de tamanho. Um invasor local ou externo poderá provocar esta falha para iniciar uma negação de serviço (DoS).

Programas Linux com o bit SUID ou SGID ativados não podem gerar, em hipótese alguma, um arquivo `core`. O administrador do Linux (e de outras variantes Unix também) pode limitar o tamanho máximo dos *core dumps*. O comando `ulimit -a | grep core` mostra os limites válidos em blocos de 512 Byte:

```
core file size (blocks, -c) 2
unlimited
```

A solução para shell scripts: quando um script iniciar um processo ele herdará o ambiente do shell em que foi chamado.

A linha `ulimit -c 0` no início do script ou num arquivo de configuração do shell bloqueia todos os arquivos `core` gerados a partir desse shell.

A solução para programas em C e o C++: a função `setrlimit()` tem no Linux a mesma função que o `ulimit` no shell (veja a **listagem 3**, função `disable_core_dumps()`).

umask

Armadilha 2: normalmente o sistema permite que todos os usuários possam ler e escrever em arquivos novos (666 em notação octal). Um usuário local, por exemplo, poderá então ler e escrever arquivos temporários e talvez até descobrir senhas. Há pouco tempo o OpenOffice foi notícia por causa do ajuste inadequado do `umask` [8]. O `umask` limita essas permissões. Muitas distribuições têm como ajuste inicial (válido para todos os usuários) `umask 022` – o que resulta em arquivos com as permissões 644. Assim, qualquer pessoa pode ver os novos arquivos, mas apenas o proprietário pode escrever neles.

A solução para shell scripts: Adicione o comando `umask 077` já no início do script para salvar o dia.

A solução para programas em C e o C++: pouco depois do início do programa, o programador chama `umask(0077)` (números representados em notação octal tem um 0 no início). Atenção: não é suficiente alterar as permissões de acesso logo depois de criar o arquivo com `chmod()`. Um invasor pode ter aberto o arquivo nesse meio tempo (a chamada *Race Condition*). Esses ataques são facilmente automatizados e surpreendentemente eficazes.

Variáveis de ambiente

Armadilha 3: variáveis de ambiente são de grande valia. Com elas o usuário configura seu shell e outros programas ou evita o uso de programas opcionais e traba-

Listagem 4: Excluindo variáveis com Zsh

```
01 function zsh_clear_env () {
02     for V in `set +`; do case "$V" in
03         '!'|'$'|'*'|'|'?'|'-'|'#'|'[0-9]|V);;
04         *) typeset +r "$V"; unset "$V";;
05     esac; done
06     unset V
07     emulate zsh
08     export IFS=" \t\n"
09     export PATH="/bin:/usr/bin:/sbin:/usr/sbin"
10 }
```

lho de digitação. Pela mesma razão elas tornam-se perigosas quando um invasor manipula seu conteúdo. São muitas as variáveis que representam um risco: `IFS`, `PATH`, `TZ`, muitas variáveis que começam com `LC_` e `LD_` e outras.

O perigo surge quando um programa confia cegamente nas variáveis de ambiente (talvez até sem saber). Por exemplo, imagine que o invasor manipulou a variável `PATH` e incluiu o diretório `/tmp` com precedência maior que qualquer outro. Depois, ele coloca um programa malicioso qualquer em `/tmp` e o renomeia para o mesmo nome de um outro programa já existente e de uso comum – `rm`, por exemplo. O usuário `root`, ignorante de tudo isso, pode tentar usar o `rm`, mas como `PATH` está manipulada, vai executar `/tmp/rm` ao invés de `/bin/rm`. Raramente fica claro qual variável está sendo usada, em que momento e de que forma, e o exemplo acima demonstra isso de maneira irrefutável.

O ambiente do processo salva as variáveis como seqüências de caracteres terminadas em zero na forma `nome=valor`. Com a função `execve()` o invasor pode ajustar qualquer coisa; por exemplo, deixando o nome da variável vazio, criar registros sem o `=` ou múltiplas versões das mesmas variáveis.

A solução para shell scripts: o *Bash* e outros shells atuais não oferecem uma solução confiável para o administrador, apenas o *Zsh* (*Z Shell*) [9] (veja a [listagem 4](#)). O programador depende do administrador de sistemas para disponibilizar um ambiente controlado para a execução de seu software. Para isso o programa Super é indicado, já que ele exclui todas as variáveis de ambiente estranhas e ajusta outras importantes com valores padrão e reconhecidamente sadios. O menos indicado é o `sudo`, porque ele apenas faz a filtragem de variáveis reconhecidamente perigosas, mas não reage às variáveis desconhecidas.

O comando `env -i` funciona bem, pois “limpa” o ambiente sem dar permissões ao programa. O comando `exec -c programa` também funciona bem. Nos dois casos o script iniciará um novo programa, aproveitando o ambiente limpo. Mas talvez nesse momento o invasor já tenha atingido seu objetivo. Para um mínimo de segurança, o script ajusta as variáveis `IFS` e `PATH` no início da execução:

```
#!/bin/sh
IFS=" \t\n"
PATH="/bin:/usr/bin:/sbin:/usr/sbin"
export IFS; export PATH
```

A solução para programas em C e C++: o programa substitui o ambiente por uma estrutura que contém apenas valores padrão. Ele apenas copia para o novo ambiente e verifica variáveis de que ele mesmo precisa. O código para exclusão total é parecido com a função `set_minimal_env()` da [listagem 3](#). ➡

Descritores abertos

Armadilha 4: o número de arquivos, de *sockets* e de outros descritores abertos é limitado. Por isso, quando estão abertos os descritores – gerados por um processo pai – são deixados de herança para os eventuais processos filhos. Um invasor pode desviar o fluxo de funcionamento de um programa executando-o com vários descritores já abertos. As possibilidades dependem muito de cada programa. Elas vão desde uso não autorizado do próprio programa e ataques de DoS até invasões sofisticadas em que o agressor consegue um shell com permissões de administrador – tudo é possível. Um processo filho também recebe todas as permissões do seu processo pai. Se, por exemplo, um programa abrir o arquivo `/etc/passwd` como usuário root e escrever nele, todo processo filho também terá permissão de escrita.

A solução para shell scripts: o `sudo` e o `Super` fecham todos os descritores de arquivos antes de chamar um comando. A chamada `exec n > arquivo` abrirá um arquivo para escrita com descritor "n" e `exec n > &-` o fechará. Para leitura o > deverá ser substituído por <.

A solução para programas em C e C++: o programa fecha os descritores já no início e depois de um `fork()` fechará todos os descritores que estiverem inesperadamente abertos (veja a **listagem 3**, função `close_descriptors()`). A função `sysconf()` com o argumento `_SC_OPEN_MAX` retorna o descritor mais alto possível no Linux (exclusivamente no Linux – outros sabores de Unix não oferecem essa possibilidade). Em outros sistemas o programador tem que contentar-se com `getdtablesize()` ou `OPEN_MAX`. Importante: `getrlimit()` não fornece o número procurado, apenas o limite para novos descritores.

A função `popen()` também chama internamente `fork()`. Como o programador não tem influência neste processo,

é melhor não usar o `popen()`. A função `execve()` fecha os descritores apenas a pedido. Uma sequência adequada é, logo depois do `open()`, o comando `fcntl(descriptor, F_SETFD, F_CLOEXEC);`.

Entrada e saída padrão

Armadilha 5: na maioria dos casos os descritores de arquivos 0, 1 e 2 são programados com a entrada e saída padrão, exceto se eles forem fechadas previamente pelo solicitante. Então, quando um programa abrir um arquivo para escrita, relacionará a ele o primeiro descritor que estiver livre, por exemplo o 2 (saída de erro padrão). A partir daí todas as mensagens de erro do sistema serão registradas – automaticamente – nesse arquivo. Como muitas vezes os invasores podem manipular essas mensagens de erro, podem usar esse artifício para alterar o conteúdo de quaisquer arquivos a seu bel-prazer – arquivos que, em condições normais, não teriam permissão para alterar.

A solução para shell scripts: os shells não escolhem automaticamente descritores adequados. No *Bash* e no *Zsh* as linhas a seguir abrem os descritores padrão quando necessário e os ligam ao dispositivo `/dev/null`:

```
test -e /dev/fd/0 || exec < /dev/null
test -e /dev/fd/1 || exec 1>&/dev/null
test -e /dev/fd/2 || exec 2>&/dev/null
```

A solução para programas em C e C++: o desenvolvedor verifica se o `stdin`, `stdout` e `stderr` estão abertos e, se necessário, faz a conexão com o `/dev/null`. Na **listagem 3**, a função `open_stdfiles()` mostra como isso funciona.

Identificadores de usuários

Armadilha 6: sistemas Unix modernos (BSD 4.4, Posix) relacionam três identificadores de usuários (UID) a cada processo. O UID efetivo (EUID) fornece as permissões de acesso, o UID real (RUID)

é o identificador do usuário que iniciou o programa. Internamente, o processo ainda possui um UID "de backup" (*Saved UID* ou *SVUID*). Em programas que ajustam o UID do usuário, o RUID e o EUID são diferentes. Em processos normais todos os UIDs são idênticos. O sistema permite, entre outras coisas, que um programa escolha qual dos três identificadores quer usar. Assim, um programa *SUID* une as permissões do usuário que chamou o programa com as permissões do proprietário do programa. O mesmo vale para os identificadores de grupos (*RGID*, *EGID* e *SGID*).

Para evitar riscos desnecessários, cada programa deveria devolver os direitos adicionais o quanto antes. O programa `passwd` necessita de permissão de root apenas para escrever uma nova senha em `/etc/shadow`, mas não quando chama ou verifica uma senha do usuário. Um bom programador cuidará de desativar irreversivelmente esses privilégios o mais cedo possível.

A solução para shell scripts: a maioria dos shells não têm a função para o script mudar sua identificação de usuário. Nesse caso o programador deve separar a tarefa em duas partes. O script privilegiado conclui seu trabalho e chama um parceiro não-privilegiado com o `Super` ou o `sudo`; o inverso também é possível.

Um script *Zsh* pode relacionar valores diretamente para as variáveis `UID` e `EUID`. Mas em função das diferenças na base entre os vários sistemas operacionais, não se deve confiar apenas em um determinado procedimento.

A solução para programas em C e C++: desativar as permissões adicionais não é fácil quando o programa tem que ser portátil. O Linux e outros sistemas Posix usam a função `setreuid()`, que altera todos os três identificadores de uma só vez (veja a **listagem 3**, função `set_credentials()`). Mas em sistemas BSD o procedimento do `setreuid()` é diferente.

```

/tmp/chroot
$ ## Installiert: sash und chroot
$ ## chroot Umgebung erzeugen
$ mkdir /tmp/chroot
$ gcc -static -o flucht flucht.c
$ cp flucht /bin/sash /tmp/chroot
$ ## chroot aktivieren
$ cd /tmp/chroot
$ chroot /tmp/chroot /sash -i
Stand-alone shell (version 3.4)
> -ls
.
flucht
sash
> ./flucht
*** Bin entkommen ***
Stand-alone shell (version 3.4)
> whoami
root
> -ls /etc/shadow*
shadow
shadow-
shadow.org
>

emacs: flucht.c
File Edit View Ctags Tools Options Functions Buffers C++ Cscope
#include <stdio.h>
#include <fcntl.h>
#include <unistd.h>
#include <sys/stat.h>

int main(void)
{
    int Fluchtweg;

    mkdir("versteck", 0755);
    Fluchtweg = open(".", O_RDONLY);
    chroot("versteck");

    /* Fluchtweg ist noch offen */
    fchdir(Fluchtweg);
    /* Bin raus aus chroot, hoch zu "/" */
    chdir(".././.././.././../");
    /* chroot rückgängig machen */
    chroot(".");

    /* Shell aufrufen */
    printf("*** Bin entkommen ***\n");
    execl("/bin/sash", "-i", NULL);
}

```

Figura 2: O pequeno programa em C no editor (à direita) serve como ajuda na fuga. Com ele o invasor (à esquerda) consegue fugir de uma jaula *chroot*

Processamento de sinais

Armadilha 7: os programas comunicam-se mais facilmente através de sinais. Quando um sinal (codificado em um valor de 5 bits) chega a um processo, o sistema operacional desencadeia uma ação. A maioria dos sinais pode ser ignorada pelo programa, se esse for o objetivo. De acordo com o caso, poderá ser mantida a reação padrão (terminar o programa, criar um *core dump*, etc) ou instalar um manipulador de sinais próprio.

O Linux distingue 31 sinais (veja a lista completa com o comando `man 7 signal`) que um processo recebe do sistema operacional, de um processo que pertence ao mesmo usuário ou ao root. Com o `SVUID` a situação fica ainda mais complexa (veja `man 2 kill`). O processo herda de seu pai o modo de reagir a um sinal. Se o processo pai tinha implementado um manipulador de sinais próprio, o processo filho retornará à reação padrão. A função `execve()`, porém, é um caso especial; veja a página de manual com: `man 2 execve`.

Utilizar os sinais de forma correta é um tanto complicado, porque durante o processamento podem surgir novas condições para os sinais. A função `signal()` também se comporta de forma diferente em variantes do Unix. Desenvolvedores e administradores pouco experientes não deveriam utilizar esta função. A probabilidade de fazer tudo da maneira certa é muito pouca. Para o invasor abrem-se três caminhos:

- ➔ O desenvolvedor de um programa com bit SUID ativado não sobregrava o UID real do solicitante. O invasor inicia então o programa e envia os sinais. O programa não esperava receber sinais de um usuário comum. O programador reage sobregravando o UID real com o UID efetivo o quanto antes (ver armadilha 6).
- ➔ O invasor solicita que o sistema operacional mande sinais que o programa não espera.

Listagem 5: sudo-clone com Gate Guardian

```

#include <stdio.h>
02 #include <stdlib.h>
03 #include <unistd.h>
04 #include "gateguardian.c"
05
06 int main(void) {
07     char pwd[9], busca[99], linha[99];
08     uid_t uid;
09     FILE *df;
10     gateg_cfg_t cfg;
11     int op;
12
13     /* Configuração padrão */
14     gateg_init_cfg(&cfg, GATEG_INIT_DFLT);
15     /* Isso só acontece na segunda chamada: */
16     op = GATEG_OP_SET | GATEG_OP_DELAY_1;
17     cfg.ops.priv_set = op;
18     cfg.ops.priv_drop = op;
19     cfg.ops.sighand_verify = op;
20     /* A primeira chamada resolve a maioria dos casos */
21     gateg_safe_init(&cfg);
22
23     /* Abre o banco de dados */
24     df = fopen("/etc/Passworte", "r+");
25     /* Por garantia, desliga os buffers */
26     setvbuf(df, 0, _IONBF, 0);
27
28     /* Lê UID e senha */
29     printf("Por faovr, informe o user ID: ");
30     fscanf(stdin, "%d", &uid);
31     printf("Por favor, informe a senha: ");
32     fscanf(stdin, "%8s", pwd);
33
34     /* Busca registro no banco de dados */
35     sprintf(busca, "%d:%s", uid, pwd);
36     while (1)
37     { /* Busca por linha */
38         if (fscanf(df, "%98s", linha) != 1)
39             exit(1); /* Fim do arquivo */
40         if (strcmp(linha, busca) == 0)
41             break; /* Encontrado */
42     }
43
44     /* Segunda chamada, substitui o setreuid */
45     cfg.ops.fds_close = op;
46     cfg.priv_uid = uid;
47     gateg_safe_init(&cfg);
48
49     execl("/bin/Skript", 0); /* Início */
50     return 255;
51 }

```

```
$ cat /tmp/ExploitSkript # Angriffs-Skript:
echo
cat 0<&3

$ cat /bin/Skript # Eigentliches Skript:
#!/bin/bash
echo -e "\n\nIch bin das bash Skript!\n"

$ # Mit BASH_ENV dem Programm unterjubeIn
$ BASH_ENV=/tmp/ExploitSkript /bin/sudo-clone

Bitte User-Id eingeben : 1000
Bitte Passwort eingeben: geheim

1001:Anna
1002:qwert123
1003:W_2d@0x_
1004:password
1005:Wald1

Ich bin das bash Skript!
$
```

Figura 3: É fácil enganar o nosso clone do sudo da listagem 2. O invasor ajusta o BASH_ENV para dar passagem a seu próprio script.

⇒ O invasor transmite ao processo filho uma forma de reação não esperada.

A solução para shell scripts: os próprios shells normalmente ajustam seus manipuladores de sinais. Mas não há garantia. Através do script, o programador não pode mudar muita coisa. O sudo e o Super ajustam o UID real correto e voltam ao processamento (e reação) padrão aos sinais.

A solução para programas em C e C++: o programa logo ignora o UID real (ver acima) e restaura os manipuladores de sinais (veja a **listagem 3**, função `reset_sighandlers()`).

Diretório de trabalho

Armadilha 8: em programas de ajuste do UID sugere-se definir um diretório de trabalho seguro; pode ser até o diretório pessoal do root (na maioria das distribuições, `/root`). Isso será importante se o diretório de trabalho estiver em uma mídia removível (ou remota, que pode estar montada ou não) ou num sistema de arquivos no qual as opções de montagem permitam acesso a outros usuários (por exemplo com `gid -option` em sistemas de arquivos FAT). Um core dump não é seguro num meio desse tipo.

Solução para shell scripts: `cd diretório || exit 1`

A solução para programas em C e C++ é o comando: `if (chdir("diretório") != 0) exit(1);`

Trocando a raiz

Armadilha 9: a função `chroot()` prende programas num diretório qualquer e “mente” a ele, dizendo que esse diretório é, na verdade, o sistema de arquivos raiz. Teoricamente isso evita o acesso ao sistema restante, já que o programa “enjaulado” tem poderes ilimitados dentro da jaula, mas permissões quase nulas fora dela. Mas a teoria, na prática, é diferente. A **figura 2** mostra como o invasor com permissões de root aproveita uma das várias brechas para sair da jaula `chroot` sem tomar conhecimento das restrições. Basta criar um arquivo e executá-lo. O usuário comum também sai sem problemas quando seu diretório de trabalho ou um outro diretório aberto está fora da jaula `chroot`.

A solução para shell scripts: usar o programa `chroot` para realizar tarefas de segurança não é recomendado. Como argumento ele recebe o comando que deve ser executado. Como isso pode ser executado apenas pelo root, o comando roda sempre com permissões de root. Consulte [9] para saber a maneira correta de fazer as coisas.

Solução para programas em C e C++: a função `safe_chroot()` da **listagem 3** mostra como o programador pode alcançar o objetivo em quatro passos.

Ataque dos clones

Voltando ao programa `sudo-clone`. Como foi dito no início, a **listagem 2** cai em pelo menos três das armadilhas que já mencionamos:

⇒ O nosso clone do sudo abre `/etc/passwd` como root, mas não fecha o arquivo antes de chamar `execl`. O `/bin/script` funciona como usuário comum, mas herda o descritor de arquivo, incluindo as permissões de leitura e escrita.

⇒ Supondo que o usuário solicitante tenha fechado a saída padrão e chame `sudo-clone`. A linha 15 atribuirá então o descritor 1 ao manipulador de arquivo `df`. O `printf()` na linha 20 vai querer abrir a entrada padrão, para isso, escreverá 1 no descritor (`stdout`). Porém, lá estará escondido o arquivo `/etc/passwd`.

⇒ Nosso clone do sudo chama o script `/bin/script`, mas não exclui previamente as variáveis de ambiente. Se um invasor ajustar a variável `BASH_ENV` em um outro script, o Bash vai executá-lo antes que `/bin/script` funcione.

Quem duvidar dessas brechas encontrará a prova da primeira e da terceira falhas na **figura 3**. O invasor usa uma variável de ambiente com a qual força o Bash a executar seu próprio script `/tmp/ExploitSkript` antes de executar o `/bin/script` solicitado. O script do exploit anexa o descritor de arquivo 3 à saída padrão (descritor 0). O descritor 3 herda o script do clone do sudo (a terceira lacuna) e recebe o arquivo da senha que chega corretamente no `stdout`.

Guardiões

Quem quiser proteger seu programa em C ou C++ destas armadilhas, sem a necessidade de cuidar de todos os detalhes,

```
$ cat /tmp/ExploitSkript # Angriffs-Skript:
echo
cat 0<&3

$ cat /bin/Skript # Eigentliches Skript:
#!/bin/bash
echo -e "\n\nIch bin das bash Skript!\n"
cat 0<&3

$ # Mit BASH_ENV dem Programm unterjubeIn
$ BASH_ENV=/tmp/ExploitSkript /bin/sudo-clone_gg

Bitte User-Id eingeben : 1000
Bitte Passwort eingeben: geheim

Ich bin das bash Skript!

/bin/Skript: 0: Bad file descriptor
$
```

Figura 4: O Gate Guardian protege o programa `sudo-clone` (**listagem 5**). O invasor não pode mais inserir seu script de exploit e o `/bin/Skript` também não herda mais o descritor de arquivo sabotado.

pode usar bibliotecas prontas. O autor deste artigo usou o *Gate Guardian* [5], distribuído sob a licença BSD). Essa biblioteca em C conhece os truques e evita a entrada dos invasores, como um discreto guardião. Só consegue entrar no sistema quem é comprovadamente inofensivo ou por um acaso encontra uma entrada desprotegida.

Para usar o Gate Guardian no próprio projeto basta incluir o cabeçalho `src/gateguardian.c` em seu programa. A **listagem 5** mostra como o desenvolvedor fecha as lacunas da **listagem 2** com a ajuda do Gate Guardian. Em casos mais simples, ele apenas anexará quatro linhas ao código-fonte. A linha inclui `gate-guardian.c` em seu programa. Isso não é muito elegante, mas evita qualquer modificação nos *Makefiles*.

A linha 10 cria uma estrutura de configuração que ajusta a função obrigatória `gateg_init_cfg()` na linha 14 com os ajustes iniciais padrão através do parâmetro `GATEG_INIT_DFLT`. Além disso existem os valores `GATEG_INIT_DFLT_PARANOID` com ajustes mais restritos e `GATEG_INIT_NOP` se o próprio programador fizer todos os ajustes. Na linha 21, em `gateg_safe_init()`, a configuração desejada é ativada. Para muitos programas isso já é o suficiente.

Quase a metade das 1600 linhas de código do Gate Guardian desvia das nove armadilhas descritas anteriormente. O código restante é responsável pelas interfaces. As medidas de segurança falam muitas vezes por falta de tempo do desenvolvedor. É para isso que o Gate Guardian Fail-Safe foi desenvolvido: ele usa os ajustes mais seguros sem medidas especiais. Isso também auxilia os administradores que precisam fazer pequenas modificações em seus servidores. Processos de instalação demorados e experiências com os *Makefiles* são proibidos. Basta copiar um arquivo e modificar algumas linhas num outro.

Os detalhes

Os comandos nas linhas **16 a 19** da **listagem 5** atribuem o valor `GATEG_OP_SET` para alguns elementos da subestrutura `ops`. Com isso cada elemento ativa uma função. Também existem `GATEG_OP_NOP` (sem ação) e `GATEG_OP_DFLT` (ação padrão, ou *default*). Há também o valor `GATEG_OP_DELAY_1`. Ele faz com que a função só opere na segunda chamada de `gate_safe_init()`.

O programa deve fechar o máximo de lacunas o quanto antes e depois fechar o resto rapidamente. Ele desistirá de suas permissões de root quando não necessitar mais delas (`priv_set` e `priv_drop` ajustarão o valor de atraso). O processo `sighand_verify` verifica se um outro usuário ainda pode mandar sinais ao processo. Esse teste também só tem sentido quando o UID final do usuário já estiver definido.

As linhas **45 a 47** dão reforço. A operação `fds_close` fecha novamente todos os descritores abertos. Isso já aconteceu no primeiro passo, mas agora um novo descritor já está aberto (`df`). Então o programa reconhece o UID do usuário procurado e o insere no elemento `priv_uid`. A linha **47** faz o restante.

A **figura 4** prova que o trabalho compensa. Tanto um ataque através do `BASH_ENV` como o ataque por um descritor não têm efeito. Para assegurar que a lacuna está mesmo trancada o comando `cat 0 &3` agora também está escrito em `/bin/script`. Isso não foi necessário na situação da **figura 3**.

Chegando ao final com segurança

Não é fácil escrever programas seguros, mas a recompensa é a paz de espírito mais tarde. Talvez as explicações acima ajudem a fazer o mundo do Software Livre respirar um pouco mais aliviado. O projeto Gate Guardian também quer contribuir com isso. Na próxima edição,

a segunda parte desta pequena série descreve o que pode acontecer de errado com arquivos se você não for atento o suficiente ao cuidar deles. ■

INFORMAÇÕES

- [1] John Viega e Matt Messier, *Secure Programming Cookbook for C and C++*, Editora O'Reilly, 2003: www.secureprogramming.com
- [2] David A. Wheeler, *Secure Programming for Linux and Unix HOWTO*: www.dwheeler.com/secure-programs
- [3] Dirk P., *Insel-Hüpfer - Sicherheitslücken bei Hosting-Providern*. Linux Magazine Alemã, edição 10/03, página 56
- [4] Steve Grubb, autor do *env_audit*: www.web-insights.net/env_audit
- [5] Página pessoal de Dominik Vogt, autor do programa *Gate Guardian*: sourceforge.net/projects/gateguardian
- [6] Página oficial do Super: freshmeat.net/projects/super
- [7] Página oficial do sudo: www.courtesan.com/sudo
- [8] Mark Vogelsberger, *InSecurity News*, Linux Magazine 11/04, página 22. Mensagem original: www.securitytracker.com/alerts/2004/Sep/1011205.html
- [9] Chroot-Login-HOWTO: www.tjw.org/chroot-login-HOWTO
- [10] Zsh: zsh.sunsite.dk
- [11] Código-fonte dos exemplos deste artigo: [ftp://ftp.linux-magazin.de/pub/listings/magazin/2005/02/Sec-Prog](http://ftp.linux-magazin.de/pub/listings/magazin/2005/02/Sec-Prog)

SOBRE O AUTOR

O matemático Dominik Vogt é desenvolvedor de software e administrador de sistemas há muitos anos. No momento trabalha como consultor autônomo de informática com foco em segurança de software. Nas suas horas livres gosta de fazer experiências com o gerenciador de janelas *Fvwm*.



BEM-VINDO À LINUX USER!

echo

```
#!/bin/bash
# subshell.sh
```

level OUTSIDE subshell

```
L" # Bash, version 3,
BASH_SUBSHELL variable.echo
if [ -z "$inner_variab
then echo "inner variabl
defined in main bod
of shell" else
echo "inner_variab
defined in main bod
```

CD do M

Para os leitores desta edição na matéria de

Trabalho

Se o painel de sedentos por KoolDock ou

Revezamento

Os operadores na Internet, e Apollon, po

Papo de

Em mais um aprender a da linha de c

exit 0

Para os leitores cansados de experimentar distribuições Linux, o CD do mês traz nesta edição uma novidade: todos os gerenciadores de conteúdo mencionados na matéria de capa desta edição, prontos para instalar.

Se o painel do KDE não tem um visual espetacular o bastante para seus olhos sedentos por beleza, você pode deixá-lo com um gostinho de Mac OS X com o *KoolDock* ou o *KXDock*.

Os operadores do *Kazaa*, a mais popular rede de compartilhamento de arquivos na Internet, só oferecem seu software para sistemas Windows; os projetos *GiFT* e *Apollon*, porém, aproximam o Kazaa do mundo Linux.

Em mais um capítulo de nossa saga através do mundo do *Shell Script*, vamos aprender a avaliar expressões, capturar sinais e receber parâmetros através da linha de comando.

Wiki, Blog, CMS e outras siglas estranhas que povoam esse exótico universo...

Sopa de letrinhas

Para os leitores cansados de experimentar distribuições Linux, o CD do mês traz nesta edição uma novidade: todos os gerenciadores de conteúdo mencionados na matéria de capa desta edição, prontos para instalar.

POR RICARDO D. DALCENO

Muito bem, você já leu as matérias de capa deste mês, está empolgado e pronto para começar o seu próprio site, mas não tem a menor idéia de qual sistema utilizar. Melhor ainda, já sabe qual deles vai utilizar, mas não quer passar horas procurando e baixando os gerenciadores de conteúdo e seus componentes. Tudo bem! Foi pensando em você que nós, da Linux Magazine, fizemos a seleção de programas que está disponível no CD-ROM deste mês.

Antes de mais nada, certifique-se de que:

- Seu servidor web está corretamente instalado e configurado.
- Seu servidor web tem suporte a PHP.
- Seu servidor possui o interpretador Python instalado e a versão é maior ou igual a 2.2.

Para saber se seu servidor web suporta PHP corretamente, crie um arquivo chamado `phptest.php` no `DocumentRoot` de seu servidor com o conteúdo do **quadro 1**. Se você tiver instalado corretamente seu servidor web e o PHP, ao abrir seu browser em <http://seu-servidor-web/phptest.php> aparecerá uma página contendo as informações sobre as configurações do PHP em seu servidor.

Para saber se sua máquina possui o interpretador Python e a qual versão ele pertence, digite em um terminal:

```
$ python -V
```

Quadro 1: phptest.php

```
<?php
    phpinfo();
?>
```

Caso precise instalar ou atualizar alguma coisa, siga as instruções de instalação (e atualização) de sua distribuição. Possivelmente, esses pacotes que você precisa instalar ou atualizar estão disponíveis no mesmo CD que você utilizou para instalar seu sistema operacional. Caso não estejam, verifique na Internet – as *urls* listadas ao final deste artigo devem levá-lo aos principais sites de que você pode precisar.

Para instalar o *Zope*, não é necessário ter um servidor web rodando, mas é aconselhável. Para instalar os “produtos” *Zope* (como são chamados os aplicativos desenvolvidos para ele), apenas descompacte o *tarball* no diretório `Products` de sua instalação *Zope*. Depois digite o comando:

```
$ <zope-home>/bin/zopectl restart
```

e pronto. Você já tem o produto (ou aplicativo) *Zope* instalado e disponível para uso. No CD, esses produtos estão disponíveis no diretório: `<cdrom>/Zope/Zope-2.7/Common`. Os arquivos de instalação do *Zope* estão disponíveis para Linux, Mac OS X (`<cdrom>/Zope/Zope-2.7/Unix`) e para Windows® (`<cdrom>/Zope/Zope-2.7/Win32`). Os produtos, por serem escritos na linguagem de programação *Python*, são independentes de plataforma. No diretório `<cdrom>/Zope/Zope-2.7/Doc` você encontrará um pequeno conjunto de documentos sobre esse servidor de aplicação. Para saber mais ou para encontrar outros produtos baseados nele, acesse o site [1].

MediaWiki, Zope, Plone Drupal, Twiki

Incluindo todos os plugins e add-ons necessários para você construir um ambiente completo de gerenciamento de conteúdo.

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 nº10 e não pode ser vendido separadamente.

Produzido no Polo Industrial de Manaus por Sonopress Rimo da Amazônia Ind. e Com. Fe

CD Número 10
julho 2005

LINUX

MAGAZINE

iki,
one,
Wiki

ons necessários
e completo de
eúdo.

gazi-
e até
ontra
tipo
ioso,
linux
iliza
erda
a da
A ou
uído.
inte-
linux
não
dido
ente.

Ind. e Com. Fonogr

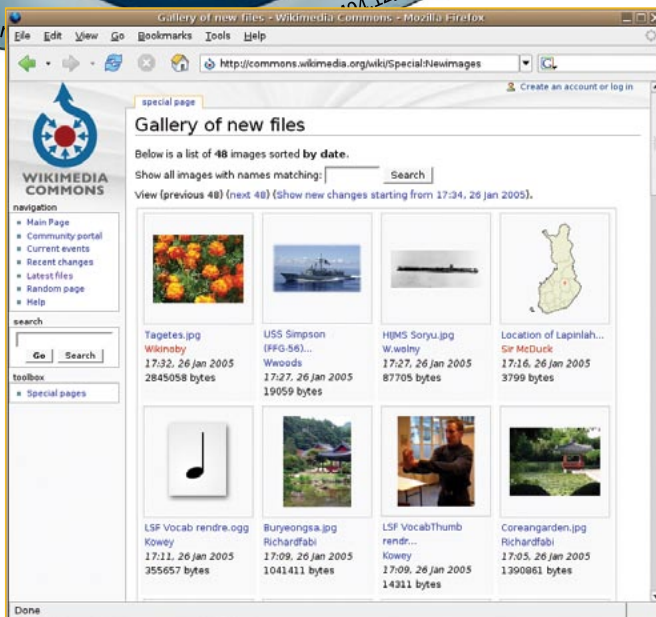


Figura 1: O Mediawiki é um dos sistemas de gerenciamento de conteúdo presentes em nosso CD.

O diretório `KDE-Addons` do CD-ROM contém dois pequenos aplicativos para o desktop KDE. Para instalar o `kxdocker`, basta descompactar o arquivo e executar o script `kxdocker.sh` em um terminal. Já o `kooldock` exige que você tenha o pacote `gcc` (e suas dependências) instalado, pois é necessário compilá-lo. Calma, **NÃO ENTRE EM PÂNICO!**. É um processo simples, fácil e rápido. Basta abrir um terminal e executar comando `./configure`; o script se encarregará de verificar se seu sistema possui tudo o que for necessário para compilar o `kooldock`. Após o término da execução do script, se não houver nenhum erro, basta digitar `su -c "make && make install"` e pronto. Viu como é simples? Se quiser saber mais sobre o processo de instalação desse software, leia os arquivos `README` e `INSTALL` disponíveis no diretório criado após descompactar o pacote. É provável que ocorra um erro de “falta de biblioteca”. Caso isso aconteça, instale as que estiverem faltando e rode novamente o script `configure` como indicado acima. Para saber quais pacotes instalar, leia a documentação da sua distribuição. Geralmente, o pacote tem o mesmo nome da biblioteca (*lib*) e pode ser seguido por um `-dev` ou `-devel`.

Para instalar o *Apollon*, cliente para redes *P2P*, basta executar, como root, o instalador encontrado em `<cdrom>/Apollon`. É necessário que o KDE esteja presente no sistema, bem como suas bibliotecas de desenvolvimento. O instalador precisa de tudo isso porque compila todos os programas a partir do código-fonte – mas o faz quase sem incomodar o usuário. Divirta-se!

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.

INFORMAÇÕES

- [1] Zope: www.zope.org/
- [2] Plone: www.plone.org/
- [3] PHP: www.php.net/
- [4] Python: www.python.org/
- [5] Apache: www.apache.org/
- [6] TWiki: www.twiki.org/
- [7] TikiWiki Brasil: br.tikiwiki.org/
- [8] Apollon: apollon.sourceforge.net/

KoolDock e KXDock

Trabalhando nas docas

Se o Kicker não tem um visual espetacular o bastante para seus olhos sedentos por beleza, você pode embonecar o KDE com o KoolDock ou o KXDock.

POR HAGEN HÖPFNER

O Kicker é o ponto de partida do KDE. Ele inicia programas, alterna entre desktops e abriga *applets* (pequenos programas que residem no painel). Além dessas funções básicas, o programa também conhece alguns truques gráficos: ele permite o uso de transparência e animações nos ícones dos atalhos, que têm um efeito de *zoom* quando o mouse passa sobre eles. Infelizmente, o painel (popularmente chamado de “barra de tarefas”) padrão do KDE não pode se igualar à elegância gráfica de sua contraparte no Mac OS X, chamada simplesmente *Dock*. Matias Fernandez, Francisco Guidi e Stefano queriam algo mais que o design padrão. Matias e Francisco desenvolveram um painel opcional para o KDE que se assemelha à Dock do Mac OS X; Stefano, para não ficar por baixo, também desenvolveu o seu. Neste artigo, descrevemos o KoolDock [1] e o KXDock [2] e revelamos como você pode enfeitar a área de trabalho do KDE com eles.

Acessórios bacanêrrimos

O KoolDock é um projeto em franco desenvolvimento dedicado a ser superbacana. Nem é preciso um compilador para ver como esse painel substituto é tudo



Figura 1: O KoolDock surge quando você arrasta o mouse para a parte de baixo da tela.

de bom. Os pacotes para Debian e Mandrake podem ser encontrados na página do projeto e os pacotes RPM para o SUSE LINUX estão disponíveis em [3]. Se você preferir compilar o programa a partir do código fonte, basta seguir os três passos de sempre: `./configure && make && su -c "make install"`. Antes de iniciar o KoolDock, aconselho a arrastar o Kicker para a direita ou a esquerda de sua tela, já que o KoolDock vai seqüestrar a parte de baixo dela. Ao digitar `kooldock`, a barra de tarefas simplesmente mostra uma mensagem mandando você mover o mouse para a parte de baixo da tela para ver o KoolDock. Essa ação faz surgir um aplicativo que se parece bastante com o Dock do Mac OS X (figura 1).

Por padrão, o KoolDock tem espaço suficiente para seis ícones de atalho para aplicativos, ou “lançadores”. Ao passar o mouse sobre os ícones, o programa aumenta o tamanho do ícone ativo. Acima dele, o KoolDock exibe a ação que realizará se você clicar com o botão esquerdo; por exemplo, na figura 1 o ícone à esquerda inicia o navegador Konqueror.

Um clique com o botão direito sobre o KoolDock abre o menu de contexto. O item *KoolDock* traz a maioria das opções que realmente importam. Por exemplo, *Go to Desktop* troca a área de trabalho em uso e *Task List* exibe uma lista das janelas ativas. *Edit Preferences* (figura 2) abre uma janela de configuração onde é possível personalizar o KoolDock.

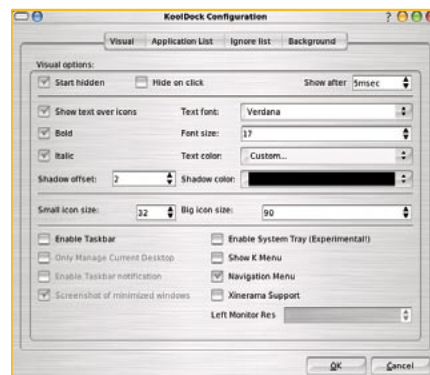


Figura 2: Use esta janela para especificar se o KoolDock deve incorporar o menu do KDE.

Mais do que chamar o menu de contexto para dar uma olhada na lista de programas ativos, vale a pena mandar o KoolDock integrar os ícones desses programas ao painel. Para isso, marque *Enable Taskbar* na aba *Visual*. Selecione *Show K Menu* para acrescentar o *Menu K* (o “Menu Iniciar” do KDE) ao KoolDock. Clique em *OK* ao terminar para adicionar um ícone para o menu no lado esquerdo do Dock. Se estiver com vontade de experimentar, tente também clicar em *Enable System Tray* para adicionar a “bandeja de sistema” (a área com os applets) do Kicker a seu KoolDock.

Sempre que você iniciar um applet do painel daqui em diante, seu ícone será exibido no KoolDock; entretanto, descobrimos que isso não funciona com todos os applets. Quando iniciamos o *kamix* (um mixer de som) à máquina de nosso laboratório para ajustar o volume no SUSE LINUX, nada aconteceu.

A aba *Application List* permite adicionar novos ícones de aplicativos ao KoolDock. O modo mais fácil de fazer isso é arrastar o lançador de um programa a partir do menu K e jogá-lo na janela de configuração. Clique em *Add* na janela seguinte para adicionar um atalho para aquele programa à sua “coleção” no KoolDock.

Se você tem o hábito de abrir uma porção de janelas de terminal ao mesmo tempo, mas não quer sacrificar todo aquele espaço no KoolDock, pode adicionar o programa do terminal à *Ignore list*. Para isso, selecione *Grab Window* e clique em uma das janelas do aplicativo que você quer remover da lista de programas ativos. Clique então em *Add* na caixa de configuração do KoolDock: isso acrescenta o programa selecionado à lista e evita que ele seja exibido como ativo no futuro.

Um toque de Mac OS X

O *KXDock* é outra barra de ícones para o KDE que se parece bastante com o Dock do Mac OS X. Os arquivos RPM do *KXDock* para Mandrake Linux podem ser instalados no Fedora Core 2 sem quaisquer problemas. No Debian e derivados (como o Ubuntu), o comando `apt-get install kxdocker` resolve o problema. Para compilar o *KXDock* a partir do código fonte, é preciso primeiro compilar o pacote *kxdocker-0.27* e em seguida *kxdocker-resources-0.9*, que contém os dados e ícones do dock.

Digite *kxdocker* para iniciar seu novo painel; novamente, o programa seqüestra a parte de baixo da tela. Desta vez, você não apenas obtém ícones de inicialização de programas, como também o menu iniciar do KDE e uma coleção de applets (figura 3). Por padrão, a seleção de applets inclui um calendário, um relógio, indi-



Figura 3: O *KXDock* já começa com um botão para o Menu K, atalhos para aplicativos e applets.

cadores de volume de tráfego na rede e carga da CPU e um mostrador do estado da bateria. Além da barra, o *KXDock* coloca um mini-ícone no painel do Kicker; clicando nele, você pode esconder ou exibir o *KXDock*.

Para remover o mostrador de estado da bateria em máquinas desktop, liberando espaço para mais lançadores de programas, selecione *Configurator* no menu de contexto do *KXDock* para acessar a caixa de configuração (figura 4). A aba *Objects* (se conseguir encontrá-la no meio de uma caixa de diálogos altamente poluída...) permite remover lançadores e applets. É só selecionar o item que deve ser removido e, no painel da direita, clicar no ícone semelhante a um círculo preto com uma cruz branca.

O ícone da varinha mágica permite criar novos lançadores de programas. Clique nela para criar um novo item e especifique suas propriedades usando as abas no lado direito da janela. O *KXDock* aceita qualquer valor para *Name*. Digite o texto a ser mostrado quando o mouse passar em *Over Text showed* e escolha um ícone pra o novo item na aba *Images*. Para rodar um programa ao clicar com o botão esquerdo num ícone, defina a ação desejada como *onClickExec* em *Actions*; poderia ser *thunderbird*, por exemplo, se você estiver criando um lançador para o cliente de email do projeto Mozilla. As setas direcionais ajudam a posicionar o novo ícone. Quanto mais para o alto da lista o ícone for movido, mais à esquerda ele aparecerá no painel. Um esquema de arrastar e soltar teria sido mais prático. Ao terminar, clique no símbolo do disquete para salvar suas alterações. Será preciso escolher um nome de arquivo e em seguida decidir se o *KXDock* deve usar essa configuração na próxima vez em que você iniciar.

O manual em [5] traz mais informações sobre a configuração do *KXDock* e inclui alguns exemplos de ajuste.

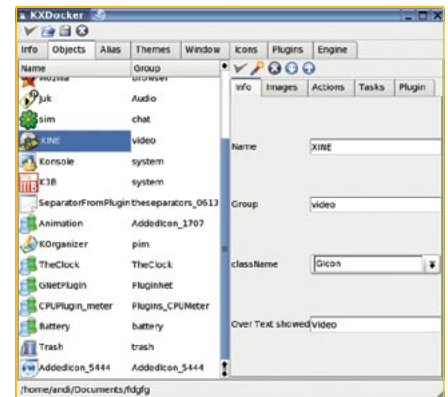


Figura 4: Nesta caixa emporcalhada configuramos o comportamento do *KXDock*.

Voluntariamente castrado

Tanto o KoolDock quanto o *KXDock* oferecem aos usuários uma barra de tarefas agradável aos olhos, mas nenhum deles pode se comparar à funcionalidade do Kicker, e sequer chegam perto da facilidade de uso e configuração do Dock do Mac OS X, seu modelo. Os desenvolvedores não se esforçaram o bastante ao integrar o menu K: um clique em seu ícone abre o menu padrão, tbastante tedioso comparado aos belos gráficos dos dois programas. Além disso, nenhum deles permite que os usuários se livrem completamente do Kicker, uma vez que ambos usam funções dele para integrar o menu iniciar do KDE.

Sem um processo do Kicker ativo, o KoolDock e o *KXDock* perdem uma boa parte de sua funcionalidade. Dito isso, se você aprecia coisinhas gráficas e está procurando um modo de dar mais vida a seu desktop, vai se divertir bastante com estes programas. ■

INFORMAÇÕES

- [1] KoolDock: ktown.kde.cl/kooldock/
- [2] KXDock: tinyurl.com/393ah
- [3] RPMs do KoolDock para o SUSE: tinyurl.com/bvuwg
- [4] Documentação do KXDock: tinyurl.com/7lmv7
- [5] RPMs do KXDock para o SUSE: tinyurl.com/dhxxq

Compartilhamento de arquivos com o Apollon e o GiFT

Revezamento olímpico

Os operadores do Kazaa, a mais popular rede de compartilhamento de arquivos na Internet, só oferecem seu software para sistemas Windows; o projeto GiFT, porém, aproxima o Kazaa do mundo Linux.

POR CARSTEN SCHNOBER

O Kazaa [1] é a mais popular rede de compartilhamento de arquivos da Internet. Até agora, o Kazaa deveu sua popularidade aos usuários de Windows® – possivelmente graças à preponderância de cópias ilegais de programas que nas lojas tem preços exorbitantes, mas entretanto inúteis para a maioria dos usuários de Linux. Além disso, o Kazaa dificulta o acesso da comunidade Linux à rede, já que o cliente ponto-a-ponto fornecido pelo fabricante é restrito aos sistemas operacionais da Microsoft.

O projeto GiFT [2] tem agora um plugin para os usuários do Linux que querem compartilhar arquivos com todo o mundo via Kazaa. O plugin implementa o protocolo de rede Fasttrack, usado pelo Kazaa e suporta, além disso, a rede de compartilhamento Gnutella [3]. Ao mesmo tempo, o protocolo OpenFasttrack oferece uma alternativa livre ao Kazaa. Há outro plugin para a rede Ares

[4], e a página do GiFT também tem caminhos para plugins do Napster [5], do Soulseek [6] e do eDonkey [7], que entretanto estão incompletos ou não funcionam mais com a versão atual. O quadro 1 descreve como adicionar plugins.

O GiFT não é um programa cliente, mas um daemon que roda em segundo plano. Quando iniciado, ele abre uma conexão às redes P2P, abrindo um gateway para programas locais. O Apollon [8], um programa para o KDE, usa o daemon do GiFT e oferece aos usuários uma interface gráfica para acesso fácil às plataformas de compartilhamento de arquivos.

Instalação

O programa Arkollon (disponível em [8]) instala o cliente gráfico Apollon e o daemon GiFT, incluindo plugins para as redes Fasttrack, Gnutella e OpenFasttrack. Como o Arkollon contém e compila o código fonte dos programas citados, você vai precisar do ambiente de desenvolvimento do KDE, além de um compilador C (gcc) e ferramentas relacionadas. A tabela 1

traz uma lista de pacotes necessários; os nomes dos pacotes podem variar entre as distribuições baseadas em RPM.

Para executar o Arkollon digite `sh nome_do_instalador`. O programa pede a senha de root e em seguida descompacta, compila e instala os programas que você selecionou (figura 1). Se estiver rodando o Ubuntu, que não tem senha de root, use o comando `sudo` em conjunto com o nome do instalador e informe sua própria senha em vez da de root. Não esqueça que, no caso do Ubuntu, seu usuário tem que estar cadastrado no arquivo `/etc/sudoers`. Depois da instalação, é só digitar `apollon`; a interface gráfica iniciará automaticamente o daemon GiFT em segundo plano. Ao rodar o programa pela primeira vez, você precisa escolher um apelido para as redes P2P e especificar as pastas que quer compartilhar com outros usuários dessas redes.

Se algum dia você decidir se livrar do Apollon e do GiFT, o Arkollon também cuida disso por você. Digite `arkollon -uninstall` para mandá-lo remover os arquivos que você instalou.

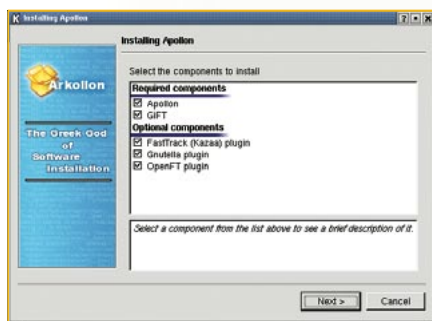


Figura 1: O Arkollon instala o Apollon automaticamente.

Quadro 1: Adicionando plugins ao GiFT

A instalação de um plugin segue sempre o mesmo padrão. Após desempacotar o arquivo com o código fonte, compile o plugin usando os comandos `./configure && make && make install` – veja na Tabela 1 os pacotes de desenvolvimento necessários para isso. A biblioteca do novo plugin é colocada no diretório `/usr/local/lib/giFT/`. No Apollon, abra então a caixa de diálogo `Settings | Configure Apollon...` e vá para a aba `Advanced`. Clique em `Add...` para abrir uma janela de navegação. Encontre `/usr/local/lib/giFT/` nessa janela e selecione o novo plugin.

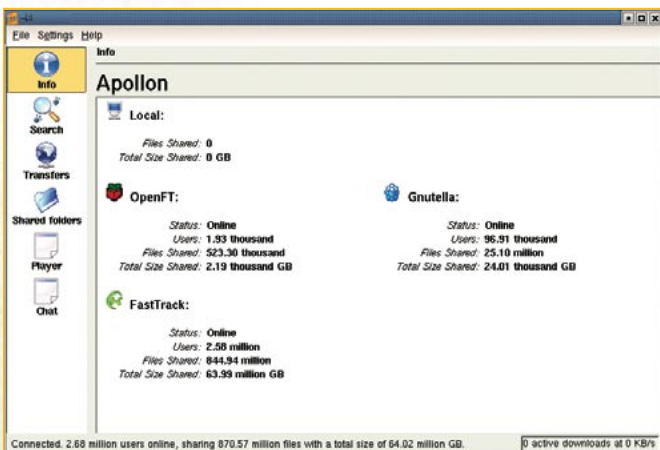


Figura 2: Muitos milhões de usuários compartilham milhares de arquivos por todo o mundo.

Ao iniciar o Apollon, o programa abre primeiro a janela *Info* mostrada na **figura 2**, que exibe o status de sua conexão a cada rede P2P. Depois de estabelecer uma conexão, vá para a janela de busca (*Search*, veja a **figura 3**) para procurar por palavras-chave entre as redes ativas. Isso preenche as áreas abaixo da caixa de busca com uma lista de arquivos encontrados. Essa lista traz o nome do arquivo e outros dados, tais como tamanho, duração ou nome do artista associado a uma música. A coluna *Availability* (disponibilidade) mostra quantas pessoas estão compartilhando esse arquivo, o que é muito importante, uma vez que, quanto mais conexões você tiver, maior será a velocidade de download.

Um clique duplo sobre um arquivo inicia o download. A janela *Transfer* traz uma visão geral dos downloads atuais. Você também pode tocar músicas de sua coleção diretamente no Apollon. Para isso, vá para a tela *Shared Folders* (*Pastas Compartilhadas*) e abra o diretório com a faixa que quer tocar. Dê um clique duplo num item para tocar a faixa; o *Player* tem os controles de reprodução.

Finalmente, o compartilhamento de arquivos!

O projeto GiFT finalmente traz aos usuários do Linux o acesso à rede do Kazaa e outras plataformas de compartilhamento de arquivos. Você pode ficar todo entu-

siasmado com o número de resultados obtidos numa busca típica, mas considere as consequências legais antes de saltar de cabeça nesse pântano (**quadro 2**). Se você baixar e oferecer conteúdo sem prestar atenção às letras miúdas, muito provavelmente vai bater de cara com a lei em algum momento – e se isso acontecer, não diga que não avisamos!

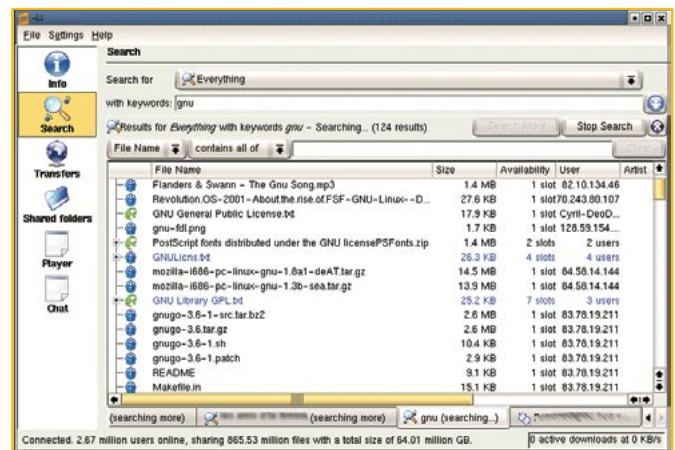


Figura 3: Qualquer busca feita retornará resultados, que podem ser muitos ou bem poucos.

Quadro 2: Legal ou ilegal?

O compartilhamento de arquivos na Internet deve sua espantosa popularidade à facilidade com que os usuários podem trocar cópias ilegais de música e software comercial. O compartilhamento de arquivos expandiu-se enormemente no final da década de 90, encabeçado pela plataforma Napster que, mais tarde, tornou-se um serviço comercial e legalizado. O algoritmo MP3 desenvolvido pelo Instituto Fraunhofer pouco antes dessa expansão foi o primeiro a permitir um formato de compressão de áudio que se prestava à troca de arquivos pela Internet. Desde então, os álbuns recém-lançados têm sido tipicamente postos à disposição para download pouco após seu lançamento, dando aos usuários acesso grátis, mas ilegal, a eles. Claro que isso também se aplica à filmes e à maior parte do software comercial.

A indústria musical em particular percebeu o perigo para seu negócio, representado por um suprimento quase ilimitado de cópias grátis de seus produtos. Mas os problemas legais foram mais ou menos resolvidos, apesar das campanhas de advertência que sugerem o contrário. As redes de compartilhamento de arquivos são basicamente legais, mas se você as usar para baixar ou propagar música ou software sem uma licença para isso, estará desobedecendo às leis de copyright. Pessoas que oferecem cópias ilegais de músicas, filmes e programas foram sentenciadas a multas de muitos milhares de dólares tanto nos Estados Unidos quanto na Europa.

Tabela 1: Pacotes necessários

RPM	Debian
<i>zlib-devel</i>	<i>zlib1g-dev</i>
<i>libtool</i>	<i>libltdl3-dev</i>
<i>db-devel</i>	<i>libdb4.2-dev</i>
<i>libmagic-devel</i>	<i>libmagic-dev</i>
<i>libvorbis-devel</i>	<i>libvorbis-dev</i>
<i>imagemagick-devel</i>	<i>libmagick-dev</i>
<i>qt3-devel</i>	<i>libqt3-mt-dev</i>
<i>qt3-devel-tools</i>	<i>qt3-dev-tools</i>
<i>kdelibs3-devel</i>	<i>kdelibs4-dev</i>

INFORMAÇÕES

- [1] Kazaa: www.kazaa.com/
- [2] GiFT: gift.sourceforge.net/
- [3] Gnutella: www.gnutella.com/
- [4] Ares: www.aresgalaxy.org/
- [5] Napster: www.napster.com/
- [6] Soulseek: www.slsknet.org/
- [7] E-Donkey: www.edonkey2000.com/
- [8] Apollon: apollon.sourceforge.net/



Dave Hamilton - www.sxc.hu

Curso de Shell Script

Papo de Botequim

Parte X

Em mais um capítulo de nossa saga através do mundo do *Shell Script*, vamos aprender a avaliar expressões, capturar sinais e receber parâmetros através da linha de comando.

POR JÚLIO CEZAR NEVES

E aê amigo, te dei a maior moleza na última aula né? Um exerciciozinho muito simples...

– É, mas nos testes que eu fiz, e de acordo com o que você ensinou sobre substituição de parâmetros, achei que deveria fazer algumas alterações nas funções que desenvolvemos para torná-las de uso geral, como você disse que todas as funções deveriam ser. Quer ver?

– Claro, né, mané, se te pedi para fazer é porque estou a fim de te ver aprender, mas peraí, dá um tempo. Chico! Manda dois, um sem colarinho! Vai, mostra aí o que você fez.

– Bem, além do que você pediu, eu reparei que o programa que chamava a função teria de ter previamente definidas a linha em que seria mostrada a mensagem e a quantidade de colunas. O que fiz foi incluir duas linhas – nas quais empreguei substituição de parâmetros – para que, caso uma dessas variáveis não fosse informada, ela recebesse um valor atribuído pela própria função. A linha de mensagem é três linhas antes do fim da tela e o total de colunas é obtido pelo comando `tput cols`. Dê uma olhada na **listagem 1** e veja como ficou:

– Gostei, você já se antecipou ao que eu ia pedir. Só pra gente encerrar esse papo de substituição de parâmetros, repare que

a legibilidade do código está “horrorível”, mas o desempenho, isto é, a velocidade de execução, está ótimo. Como funções são coisas muito pessoais, já que cada um usa as suas e quase não há necessidade de manutenção, eu sempre opto pelo desempenho.

Hoje vamos sair daquela chatura que foi o nosso último papo e voltar à lógica, saindo da decoreba. Mas volto a te lembrar: tudo que eu te mostrei da última vez aqui no Boteco do Chico é válido e quebra um galhão. Guarde aqueles guardanapos que rabiscamos porque, mais cedo ou mais tarde, eles lhe vão ser muito úteis.

O comando eval

Vou te dar um problema que eu duvido que você resolva:

```
$ var1=3
$ var2=var1
```

Te dei essas duas variáveis e quero que você me diga como eu posso, me referindo apenas à variável `var2`, listar o valor de `var1` (que, no nosso caso, é 3).

– Ah, isso é mole, mole! É só digitar esse comando aqui:

```
echo `echo $var2`
```

Listagem 1: função 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 $3=n, a linha
06 # abaixo colocaria em Msg o valor "Aceita? (S/n)"
07 TotCols=${TotCols:-$(tput cols)} # Se não estava definido, agora está
08 LinhaMsg=${LinhaMsg:-$((($tput lines)-3))} # Idem
09 Msg="$1 (`echo $2 | tr a-z A-Z`/`echo $3 | tr A-Z a-z`)"
10 TamMsg=${#Msg}
11 Col=$((TotCols - TamMsg) / 2) # Para centralizar Msg na linha
12 tput cup $LinhaMsg $Col
13 read -n1 -p "$Msg " SN
14 SN=${SN:-$2} # Se vazia coloca o padrão em SN
15 SN=$(echo $SN | tr A-Z a-z) # A saída de SN será em minúsculas
16 tput cup $LinhaMsg $Col; tput el # Apaga Msg da tela
```


Repare que eu coloquei o `echo $var2` entre crases (`), porque dessa forma ele terá prioridade de execução e resultará em `var1`. E `echo $var1` produzirá 3...

– Ah, é? Então execute para ver se está correto.

```
$ echo ``echo $var2`
$var1
```

– Ué! Que foi que aconteceu? O meu raciocínio me parecia bastante lógico...

– O seu raciocínio realmente foi lógico, o problema é que você esqueceu de uma das primeiras coisas de que te falei aqui no Boteco e que vou repetir. O Shell usa a seguinte ordem para resolver uma linha de comando:

- ⇒ Resolve os redirecionamentos;
- ⇒ Substitui as variáveis pelos seus valores;
- ⇒ Resolve e substitui os meta caracteres;
- ⇒ Passa a linha já toda esmiuçada para execução.

Dessa forma, quando o interpretador chegou na fase de resolução de variáveis, que como eu disse é anterior à execução, a única variável existente era `var2` e por isso a tua solução produziu como saída `$var1`. O comando `echo` identificou isso como uma cadeia de caracteres e não como uma variável.

Problemas desse tipo são relativamente freqüentes e seriam insolúveis caso não existisse a instrução `eval`, cuja sintaxe é `eval cmd`, onde `cmd` é uma linha de comando qualquer, que você poderia inclusive executar direto no prompt do terminal. Quando você põe o `eval` na frente, no entanto, o que ocorre é que o Shell trata `cmd` como um parâmetro do `eval` e, em seguida, o `eval` executa a linha recebida, submetendo-a ao Shell. Ou seja, na prática `cmd` é analisado duas vezes. Dessa forma, se executássemos o comando que você propôs colocando o `eval` na frente, teríamos a saída esperada. Veja:

```
$ eval echo ``echo $var2`
3
```

Esse exemplo também poderia ter sido feito de outra maneira. Dá só uma olhada:

```
$ eval echo \$$var2
3
```

Na primeira passada a contrabarra (\) seria retirada e `$var2` seria resolvido produzindo `var1`. Na segunda passada teria sobrado `echo $var1`, que produziria o resultado esperado. Agora vou colocar um comando dentro de `var2` e executar:

```
$ var2=ls
$ $var2
10porpag1.sh alo2.sh incusu logado
10porpag2.sh ArqDoDOS.txt1 listamusica logaute.sh
10porpag3.sh confuso listartista mandamsg.func
alo1.sh contpal.sh listartista3 monbg.sh
```

Agora vamos colocar em `var2` o seguinte: `ls $var1`; e em `var1` vamos colocar `1*`, vejamos o resultado:

```
$ var2='ls $var1'
$ var1='1*'
$ $var2
ls: $var1: No such file or directory
$ eval $var2
listamusica listartista listartista3 logado logaute.sh
```

Novamente, no tempo de substituição das variáveis, `$var1` ainda não havia se apresentado ao Shell para ser resolvida. Assim, só nos resta executar o comando `eval` para dar as duas passadas necessárias.

Uma vez um colega da excelente lista de discussão groups.yahoo.com/group/shell-script colocou uma dúvida: queria fazer um menu que numerasse e listasse todos os arquivos com extensão `.sh` e, quando o operador escolhesse uma opção, o programa correspondente fosse executado. Veja minha proposta na **listagem 2**:

Listagem 2: fazmenu.sh

```
01 #!/bin/bash
02 #
03 # Lista que enumera os programas com extensão .sh no
04 # diretório corrente e executa o escolhido pelo operador
05 #
06 clear; i=1
07 printf "%11s\t%s\n" Opção Programa
08 CASE='case $opt in'
09 for arq in *.sh
10 do
11     printf "\t%03d\t%s\n" $i $arq
12     CASE="$CASE
13     \"$(printf \"%03d\")\t %s;\" $i $arq)
14     i=$((i+1))
15 done
16 CASE="$CASE
17     *) . erro;;
18 esac"
19 read -n3 -p "Informe a opção desejada: " opt
20 echo
21 eval "$CASE"
```

Parece complicado porque usei muitos `printf` para formatação da tela, mas na verdade é bastante simples: o primeiro `printf` foi colocado para imprimir o cabeçalho e logo em seguida comecei a montar dinamicamente a variável `$CASE`, na qual ao final será feito um eval para execução do programa escolhido. Repare no entanto que dentro do loop do `for` existem dois `printf`: o primeiro serve para formatar a tela e o segundo para montar o `case` (se antes do comando `read` você colocar uma linha `echo "$CASE"`, verá que o comando `case` montado dentro da variável está todo indented. Frescura, né?:). Na saída do `for`, foi adicionada uma linha à variável `$CASE` para, no caso de uma escolha inválida, ser executada uma função externa para exibir mensagens de erro. Vamos executar o script para ver a saída gerada:

```
$ fazmenu.sh
  Opcao      Programa
  001        10porpag1.sh
  002        10porpag2.sh
  003        10porpag3.sh
  004        alo1.sh
  005        alo2.sh
  006        contpal.sh
  007        fazmenu.sh
  008        logaute.sh
  009        monbg.sh
  010        readpipe.sh
  011        redirread.sh
Informe a opção desejada:
```

Seria interessante incluir uma opção para terminar o programa e, para isso, seria necessária a inclusão de uma linha após o loop de montagem da tela e a alteração da linha na qual fazemos a atribuição final do valor da variável `$CASE`. Veja na **listagem 3** como ele ficaria:

Existe no Linux uma coisa chamada sinal (`signal`). Existem diversos sinais que podem ser mandados para (ou gera-

dos por) processos em execução. Vamos, de agora em diante, dar uma olhadinha nos sinais enviados aos processos e mais à frente vamos dar uma passada rápida pelos sinais gerados pelos processos. Para mandar um sinal a um processo, usamos normalmente o comando `kill`, cuja sintaxe é:

```
$ kill -sig PID
```

Onde `PID` é o identificador do processo (*Process Identification* ou *Process ID*). Além do comando `kill`, algumas seqüências de teclas também podem gerar sinais. A **tabela 1** mostra os sinais mais importantes para monitorarmos:

Além desses, existe o famigerado sinal `-9` ou `SIGKILL` que, para o processo que o está recebendo, equivale a meter o dedo no botão de desligar do computador – o que é altamente indesejável, já que muitos programas necessitam

"limpar a área" ao seu término. Se seu encerramento ocorrer de forma prevista, ou seja, se tiver um término normal, é muito fácil fazer essa limpeza; porém, se o seu programa tiver um fim brusco, muita coisa ruim pode ocorrer:

- ⇒ É possível que em um determinado espaço de tempo, o seu computador esteja cheio de arquivos de trabalho inúteis
- ⇒ Seu processador poderá ficar atolado de processos *zombies* e *defuncts* gerados por processos filhos que perderam os pais e estão "órfãos";
- ⇒ É necessário liberar *sockets* abertos para não deixar os clientes congelados;
- ⇒ Seus bancos de dados poderão ficar corrompidos porque sistemas gerenciadores de bancos de dados necessitam de um tempo para gravar seus buffers em disco (`commit`).

Enfim, existem mil razões para não usar um `kill` com o sinal `-9` e para monitorar o encerramento anormal de programas.

Listagem 3: Nova versão do fazmenu.sh

```
01 #!/bin/bash
02 #
03 # Lista enumerando os programas com extensão .sh no
04 # diretório corrente; executa o escolhido pelo operador
05 #
06 clear; i=1
07 printf "%11s\t%s\n\n" Opção Programa
08 CASE='case $opt in'
09 for arq in *.sh
10 do
11     printf "\t%03d\t%s\n" $i $arq
12     CASE="$CASE
13     "$(printf "%03d\t\t%s;" $i $arq)
14     i=$((i+1))
15 done
16 printf "\t%d\t%s\n\n" 999 "Fim do programa" # Linha incluída
17 CASE="$CASE
18     999)      exit;;                # Linha alterada
19     *)        ./erro;;
20 esac"
21 read -n3 -p "Informe a opção desejada: " opt
22 echo
23 eval "$CASE"
```

O comando trap

Para fazer a monitoração de sinais existe o comando `trap`, cuja sintaxe pode ser uma das mostradas a seguir:

```
trap "cmd1; cmd2; cmdn" S1 S2 ... SN
trap 'cmd1; cmd2; cmdn' S1 S2 ... SN
```

Onde os comandos `cmd1`, `cmd2`, `cmdn` serão executados caso o programa receba os sinais `S1`, `S2` ... `SN`. As aspas (") ou as apóstrofes (') só são necessárias caso o `trap` possua mais de um comando `cmd` associado. Cada uma delas pode ser também uma função interna, uma externa ou outro script.

Para entender o uso de aspas (") e apóstrofes (') vamos recorrer a um exemplo que trata um fragmento de um script que faz uma transferência de arquivos via FTP para uma máquina remota (`$RemoComp`), na qual o usuário é `$Fulano`, sua senha é `$Segredo` e o arquivo a ser enviado é `$Arq`. Suponha ainda que essas quatro variáveis foram recebidas por uma rotina anterior de leitura e que esse script seja muito usado por diversas pessoas. Vejamos o trecho de código a seguir:

```
ftp -ivn $RemoComp << FimFTP >> /tmp/$$ 2
2> /tmp/$$
user $Fulano $Segredo
binary
get $Arq
FimFTP
```

Repare que tanto as saídas dos diálogos do FTP como os erros encontrados estão sendo redirecionados para `/tmp/$$`, o que é uma construção bastante comum para arquivos temporários usados em scripts com mais de um usuário, porque `$$` é a variável que contém o número do processo (PID), que é único. Com esse tipo de construção evita-se que dois ou mais usuários disputem a posse e os direitos sobre um arquivo.

Caso a transferência seja interrompida por um `kill` ou um `[CTRL]+[C]`, certamente deixará lixo no disco. É exatamente essa a forma mais comum de uso do comando `trap`. Como isso é trecho de um script devemos, logo no início dele, digitar o comando:

```
trap 'rm -f /tmp/$$ ; exit' 0 1 2 3 15
```

Dessa forma, caso houvesse uma interrupção brusca (sinais 1, 2, 3 ou 15) antes do programa encerrar (no `exit` dentro do comando `trap`), ou um fim normal (sinal 0), o arquivo `/tmp/$$` seria removido.

Caso não houvesse a instrução `exit` na linha de comando do `trap`, ao final da execução dessa linha o fluxo do programa retornaria ao ponto em que estava quando recebeu o sinal que originou a execução desse `trap`.

Note também que o Shell pesquisa a linha de comando uma vez quando o `trap` é interpretado (e é por isso que é usual colocá-lo no início do programa) e novamente quando um dos sinais listados é recebido. Então, no último exemplo, o valor de `$$` será substituído no momento em que o comando `trap` é lido pela primeira vez, já que as aspas (") não protegem o cifrão (\$) da interpretação do Shell.

Se você quisesse fazer a substituição somente ao receber o sinal, o comando deveria ser colocado entre apóstrofes ('). Assim, na primeira interpretação do `trap`, o Shell não veria o cifrão (\$), as apóstrofes (') seriam removidas e, finalmente, o Shell poderia substituir o valor da variável. Nesse caso, a linha ficaria assim:

```
trap 'rm -f /tmp/$$ ; exit' 0 1 2 3 15
```

Suponha dois casos: você tem dois scripts que chamaremos de `script1`, cuja primeira linha será um `trap`, e `script2`, colocado em execução por `script1`. Por serem dois processos diferentes, terão dois PIDs distintos.

Tabela 1: Principais sinais

Código	Nome	Gerado por:
0	EXIT	Fim normal do programa
1	SIGHUP	Quando o programa recebe um <code>kill -HUP</code>
2	SIGINT	Interrupção pelo teclado. (<code>[CTRL]+[C]</code>)
3	SIGQUIT	Interrupção pelo teclado (<code>[CTRL]+[I]</code>)
15	SIGTERM	Quando o programa recebe um <code>kill</code> ou <code>kill -TERM</code>

1º Caso: O comando `ftp` encontra-se em `script1`. Nesse caso, o argumento do comando `trap` deveria vir entre aspas (") porque, caso ocorresse uma interrupção (`[CTRL]+[C]` ou `[CTRL]+[I]`) no `script2`, a linha só seria interpretada nesse momento e o PID do `script2` seria diferente do encontrado em `/tmp/$$` (não esqueça que `$$` é a variável que contém o PID do processo ativo);

2º Caso: O comando `ftp` encontra-se em `script2`. Nesse caso, o argumento do comando `trap` deveria estar entre apóstrofes ('), pois caso a interrupção se desse durante a execução de `script1`, o arquivo não teria sido criado; caso ela ocorresse durante a execução de `script2`, o valor de `$$` seria o PID desse processo, que coincidiria com o de `/tmp/$$`.

O comando `trap`, quando executado sem argumentos, lista os sinais que estão sendo monitorados no ambiente, bem como a linha de comando que será executada quando tais sinais forem recebidos. Se a linha de comandos do `trap` for nula (vazia), isso significa que os sinais especificados devem ser ignorados quando recebidos. Por exemplo, o comando `trap "" 2` especifica que o sinal de interrupção (`[CTRL]+[C]`) deve ser ignorado. No último exemplo, note que o primeiro argumento deve ser especificado para que o sinal seja ignorado e não é equivalente a escrever `trap 2`, cuja finalidade é retornar o sinal 2 ao seu estado padrão. ➔

Se você ignorar um sinal, todos os sub-shells irão ignorá-lo. Portanto, se você especificar qual ação deve ser tomada quando receber um sinal, todos os sub-shells irão tomar a mesma ação quando receberem esse sinal. Ou seja, os sinais são automaticamente exportados. Para o sinal mostrado (sinal 2), isso significa que os sub-shells serão encerrados. Suponha que você execute o comando `trap "" 2` e então execute um sub-shell, que tornará a executar outro script como um sub-shell. Se for gerado um sinal de interrupção, este não terá efeito nem sobre o Shell principal nem sobre os sub-shell por ele chamados, já que todos eles ignorarão o sinal.

Em korn shell (*ksh*) não existe a opção `-s` do comando `read` para ler uma senha. O que costumamos fazer é usar o comando `stty` com a opção `-echo`, que inibe a escrita na tela até que se encontre um `stty echo` para restaurar essa escrita. Então, se estivéssemos usando o interpretador *ksh*, a leitura da senha teria que ser feita da seguinte forma:

```
echo -n "Senha: "
stty -echo
read Senha
stty echo
```

O problema com esse tipo de construção é que, caso o operador não soubesse a senha, ele provavelmente teclaria `[CTRL]+[C]` ou um `[CTRL]+[N]` durante a instrução `read` para descontinuar o programa e, caso agisse dessa forma, o seu terminal estaria sem `echo`. Para evitar que isso aconteça, o melhor a fazer é:

```
echo -n "Senha: "
trap "stty echo
      exit" 2 3
stty -echo
read Senha
stty echo
trap 2 3
```

Para terminar esse assunto, abra um console gráfico e escreva no prompt de comando o seguinte:

```
$ trap "echo Mudou o tamanho da janela" 28
```

Em seguida, pegue o mouse e arraste-o de forma a variar o tamanho da janela corrente. Surpreso? É o Shell orientado a eventos... Mais unzinho, porque não consigo resistir. Escreva isto:

```
$ trap "echo já era" 17
```

Em seguida digite:

```
$ sleep 3 &
```

Você acabou de criar um sub-shell que irá dormir durante três segundos em background. Ao fim desse tempo, você receberá a mensagem "já era", porque o sinal 17 é emitido a cada vez em que um sub-shell termina a sua execução. Para devolver esses sinais ao seu comportamento padrão, digite: `trap 17 28`.

Muito legal esse comando, né? Se você descobrir algum material bacana sobre uso de sinais, por favor me informe por email, porque é muito rara a literatura sobre o assunto.

Comando `getopts`

O comando `getopts` recupera as opções e seus argumentos de uma lista de parâmetros de acordo com a sintaxe POSIX.2, isto é, letras (ou números) após um sinal de menos (-) seguidas ou não de um argumento; no caso de somente letras (ou números), elas podem ser agrupadas. Você deve usar esse comando para "fatiar" opções e argumentos passados para o seu script.

A sintaxe é `getopts cadeiaadeopcoes nome`. A `cadeiadeopcoes` deve explicitar uma cadeia de caracteres com todas as opções reconhecidas pelo script; assim, se ele reconhece as opções `-a`, `-b` e `-c`,

`cadeiadeopcoes` deve ser `abc`. Se você desejar que uma opção seja seguida por um argumento, ponha um sinal de dois pontos (:) depois da letra, como em `a:bc`. Isso diz ao `getopts` que a opção `-a` tem a forma `-a argumento`. Normalmente um ou mais espaços em branco separam o parâmetro da opção; no entanto, `getopts` também manipula parâmetros que vêm colados à opção como em `-aargumento`. `cadeiadeopcoes` não pode conter um sinal de interrogação (?).

O nome constante da linha de sintaxe acima define uma variável que receberá, a cada vez que o comando `getopts` for executado, o próximo dos parâmetros posicionais e o colocará na variável `nome`. `getopts` coloca uma interrogação (?) na variável definida em `nome` se achar uma opção não definida em `cadeiadeopcoes` ou se não achar o argumento esperado para uma determinada opção.

Como já sabemos, cada opção passada por uma linha de comandos tem um índice numérico; assim, a primeira opção estará contida em `$1`, a segunda em `$2` e assim por diante. Quando o `getopts` obtém uma opção, ele armazena o índice do próximo parâmetro a ser processado na variável `OPTIND`.

Quando uma opção tem um argumento associado (indicado pelo : na `cadeiadeopcoes`), `getopts` armazena o argumento na variável `OPTARG`. Se uma opção não possuir argumento ou se o argumento esperado não for encontrado, a variável `OPTARG` será "apagada" (com `unset`). O comando encerra sua execução quando:

- ⇒ Encontra um parâmetro que não começa com um hífen (-).
- ⇒ O parâmetro especial `--` indica o fim das opções.
- ⇒ Quando encontra um erro (por exemplo, uma opção não reconhecida).

O exemplo da **listagem 4** é meramente didático, servindo para mostrar, em um pequeno fragmento de código, o uso pleno do comando.

Para entender melhor, vamos executar o script:

```
$ getoptst.sh -h -Pimpressora arq1 arq2
getopts fez a variavel OPT_LETRA igual a 'h'
    OPTARG eh ''
getopts fez a variavel OPT_LETRA igual a 'P'
    OPTARG eh 'impressora'
Dispensando os primeiros $OPTIND-1 = 2 argumentos
O que sobrou da linha de comandos foi 'arq1 arq2'
```

Dessa forma, sem ter muito trabalho, separei todas as opções com seus respectivos argumentos, deixando somente os parâmetros que foram passados pelo operador para posterior tratamento. Repare que, se tivéssemos escrito a linha de comando com o argumento (*impressora*) separado da opção (*-P*), o resultado seria exatamente o mesmo, exceto pelo *OPTIND*, já que nesse caso ele identifica um conjunto de três opções (ou argumentos) e, no anterior, somente dois. Veja só:

```
$ getoptst.sh -h -P impressora arq1 arq2
getopts fez a variavel OPT_LETRA igual a 'h'
    OPTARG eh ''
getopts fez a variavel OPT_LETRA igual a 'P'
```

Listagem 4: getoptst.sh

```
01 $ cat getoptst.sh
02 #!/bin/sh
03
04 # Execute assim:
05 #
06 #     getoptst.sh -h -Pimpressora arq1 arq2
07 #
08 # e note que as informações de todas as opções são exibidas
09 #
10 # A cadeia 'P:h' diz que a opção -P é uma opção complexa
11 # e requer um argumento e que h é uma opção simples que não requer
12 # argumentos.
13
14 while getopts 'P:h' OPT_LETRA
15 do
16     echo "getopts fez a variavel OPT_LETRA igual a '$OPT_LETRA'"
17     echo "    OPTARG eh '$OPTARG'"
18 done
19 used_up=`expr $OPTIND - 1`
20 echo "Dispensando os primeiros \ $OPTIND-1 = $used_up argumentos"
21 shift $used_up
22 echo "O que sobrou da linha de comandos foi '$*'"
```

```
    OPTARG eh 'impressora'
Dispensando os primeiros $OPTIND-1 = 3 argumentos
O que sobrou da linha de comandos foi 'arq1 arq2'
```

Repare, no exemplo a seguir, que se passarmos uma opção inválida a variável *\$OPT_LETRA* receberá um ponto de interrogação (?) e a *\$OPTARG* será "apagada" (*unset*).

```
$ getoptst.sh -f -Pimpressora arq1 arq2 # A opção -f não é valida
./getoptst.sh: illegal option -- f
getopts fez a variavel OPT_LETRA igual a '?'
    OPTARG eh ''
getopts fez a variavel OPT_LETRA igual a 'P'
    OPTARG eh 'impressora'
Dispensando os primeiros $OPTIND-1 = 2 argumentos
O que sobrou da linha de comandos foi 'arq1 arq2'
```

– Me diz uma coisa: você não poderia ter usado um condicional com *case* para evitar o *getopts*?

– Poderia sim, mas para quê? Os comandos estão aí para serem usados... O exemplo foi didático, mas imagine um programa que aceitasse muitas opções e cujos parâmetros poderiam ou não estar colados às opções, sendo que as opções também poderiam ou não estar coladas: ia ser um *case* infernal! Com *getopts*, é só seguir os passos acima.

– É... Vendo dessa forma, acho que você tem razão. É porque eu já estou meio cansado com tanta informação nova na minha cabeça. Vamos tomar a saideira ou você ainda quer explicar alguma particularidade do Shell?

– Nem um nem outro, eu também já cansei mas hoje não vou tomar a saideira porque estou indo dar aula na UniRIO, que é a primeira universidade federal que está preparando seus alunos do curso de graduação em Informática para o uso de Software Livre. Mas antes vou te deixar um problema para te encucar: quando você varia o tamanho de uma janela do terminal, no centro dela não aparece dinamicamente, em vídeo reverso, a quantidade de linhas e colunas? Então! Eu quero que você reproduza isso usando a linguagem Shell. Chico, traz rapidinho a minha conta! Vou contar até um e se você não trouxer eu me mando!

Não se esqueça, qualquer dúvida ou falta de companhia para um chope é só mandar um email para julio.neves@gmail.com. Vou aproveitar também para mandar o meu jabá: diga para os amigos que quem estiver a fim de fazer um curso porreta de programação em Shell que mande um e-mail para julio.neves@tecnohall.com.br para informar-se. Valeu!

Grandes projetos de uma Sociedade Livre

Imagem original: NevraX Design Team



Vamos voltar um pouco na história e lembrar alguns passos muito importantes para a criação da cybercultura que temos hoje, como a ARPANET e o início da Internet.

POR CHRISTIANO ANDERSON

O crescimento da Internet está levando a humanidade a criar a uma nova cultura, um novo mundo. Hoje em dia é muito difícil encontrar alguém que não possua um endereço de email ou que nunca tenha navegado por páginas web. As empresas adotaram a Internet como principal meio de comunicação; o email e comunicadores instantâneos como o *Jabber* [1] são ferramentas indispensáveis para agilizar o trabalho e diminuir custos com ligações telefônicas. Estamos entrando em uma realidade em que até o telefone está sendo substituído pela tecnologia *VoIP* (*Voice Over IP*, ou Voz sobre IP. Veja mais sobre o tema na terceira edição da *Linux Magazine Brasil*).

A ARPANET [2] surgiu em 1969 e a primeira mensagem eletrônica de que se tem registro foi enviada por essa rede em 29 de outubro do mesmo ano. Apesar de a rede ter sido restrita apenas a centros de pesquisa e universidades, esse foi o início da *cybercultura*. Pesquisadores separados por milhares de quilômetros começaram a discutir em conjunto o crescimento dessa rede, que acabou por dar origem à Internet, uma rede global e independente. A maioria dos protocolos que conhecemos, como o HTTP, o SMTP e o POP tiveram seus rascunhos feitos durante a era ARPANET e foram concretizados e colocados em prática durante a Internet. Com a definição aberta desses protocolos, muitas empresas se

aproveitaram para criar softwares para funcionar nessa rede, como servidores, programas para envio de mensagens (email) e murais eletrônicos (BBS), entre outros. O primeiro SPAM [3] de que se tem registro também surgiu durante a ARPANET e foi enviado pela empresa *Digital Equipment Corporation* (DEC) anunciando o lançamento de novos produtos. A reação de repúdio foi imediata e os membros da ARPANET enviaram várias respostas à DEC tentando coibir aquele ato. Um desses membros foi Richard Stallman, que em 8 de maio de 1978 enviou sua mensagem comentando o incidente (veja referência [3]), embora sua reação inicial não tenha sido a condenação do ato.

Várias empresas iniciaram o desenvolvimento de softwares para Internet, em sua maioria softwares proprietários. A resposta da comunidade foi imediata e alguns grupos de desenvolvedores independentes se uniram para compartilhar seus conhecimentos e desenvolver software livre. Nasceram então inúmeros esforços comunitários como o projeto GNU, o kernel Linux e projeto Debian GNU/Linux [4], entre outros. Com o desenvolvimento colaborativo é possível atingir metas de maneira muito eficiente, algo que nenhuma empresa jamais conseguirá acompanhar. Isso acontece porque o desenvolvimento ocorre de maneira ininterrupta, ou seja, quando um programador finaliza seu dia de trabalho,

outro em uma parte distante do mundo inicia sua jornada de desenvolvimento, graças ao fuso horário.

Um exemplo prático: um desenvolvedor Linux que mora em São Paulo inicia sua jornada às 9 horas da manhã. Esse desenvolvedor aplica vários patches a um determinado pacote e, às 21 horas, encerra sua jornada, enviando suas contribuições e patches para o servidor CVS do projeto em que trabalha. Devido ao fuso horário, nesse exato momento um outro desenvolvedor no Japão inicia seu turno, sincroniza seu repositório local com o servidor CVS e encontra o trabalho feito pelo brasileiro. O japonês faz então todos os testes e continua o desenvolvimento do brasileiro, finalizando também às 21 horas do horário japonês. Nisto o brasileiro está prestes a iniciar novamente sua jornada, e terá seu trabalho adiantado e poderá se beneficiar das contribuições de seu colega japonês. Nenhuma empresa jamais conseguirá ter um ciclo de desenvolvimento tão rápido quanto o conseguido graças à forma colaborativa que encontramos hoje no Software Livre. ■

INFORMAÇÕES

[1] www.jabber.org

[2] pt.wikipedia.org/wiki/Arpanet

[3] www.templetons.com/brad/spamreact.html

[4] www.debian.org

Eventos agitam comunidade e mercado de TI

Centro das atenções

O Software Livre “está na boca do povo”. Prova disso são os diversos eventos que nos últimos meses agitaram a comunidade e o mercado de TI no Brasil.

POR RAFAEL PEREGRINO DA SILVA

Os últimos meses foram pródigos em eventos em que o Linux foi, se não o principal, um dos temas que mais atraiu a atenção de seus visitantes. De eventos especializados, passando por eventos da comunidade voltados para grupos de usuários até eventos governamentais, todo mundo quer saber o que a baiana, digo, o pingüim tem. E isso não é sem motivo: quaisquer que sejam os cenários de aplicação, qualquer que seja a finalidade, o Linux e o Software Livre trazem na bagagem promessas de economia e otimização de recursos, controle estratégico e independência, segurança e estabilidade. Quem é que pode resistir?

III Encontro Nacional LinuxChix Brasil – a Expedição

Belo Horizonte sediou o III Encontro Nacional LinuxChix Brasil (www.linuxchix.org.br), apelidado de “A expedição”, ocorrido entre os dias 30 de abril e 1º de maio.

Mantendo a tradição de ser um evento técnico, que preza pela objetividade, as palestras desenvolvidas foram práticas e de alto nível, dirigidas a um público de mais de 350 pessoas. Como de costume, Piter Punk abriu o evento com uma palestra sobre Slackware, desta vez ensinando a portar a distribuição para outras arquiteturas. Em homenagem ao LinuxChix, Piter Punk trajava uma das saias de sua esposa e, segundo os participantes, era “a chix” mais *fashion* do evento. Na sequência, Júlio Neves fez um relato da história do desenvolvimento de software brasileiro, com a palestra *Da reserva de mercado ao Software Livre*, mostrando o quanto o Brasil já foi importante



Figura 1: Konqui observa uma das meninas da organização, que não descansaram um segundo sequer durante o III Encontro Nacional LinuxChix Brasil em Belo Horizonte.

no cenário mundial de desenvolvimento de software. Depois dele, Fabianne Balvedi apresentou o projeto *Estúdio Livre – ilhas de produção de áudio e vídeo baseadas em Software Livre*. Em paralelo, Gustavo Noronha (o “Kov”) ensinava a uma turma curiosa como trabalhar com o *Subversion*, uma alternativa para o CVS – isso para citar apenas algumas das palestras iniciais.

Outros destaques foram o curso de *Animação em Software Livre*, a cargo de Fabianne Balvedi e Rodrigo Domingues (o “Spy”), e a palestra de Rodrigo Rubira Branco, sobre *Segurança e Kernel*, onde apresentou técnicas para modificar o kernel do sistema de modo a aumentar o seu nível de segurança. Ao final do primeiro dia, Sulamita Garcia apresentou as diferenças



Telefonia IP

Com Software Livre

entre as distribuições que originaram a grande maioria das centenas de distribuições existentes hoje em dia. Ao final da apresentação, houve um pequeno e inevitável debate sobre o modelo de desenvolvimento do Debian e a esperança de que o novo líder do projeto ponha em prática um modelo de desenvolvimento mais dinâmico.

O segundo dia começou com palestras de Gustavo Noronha, desenvolvedor do Gnome, e Hélio Castro, desenvolvedor do KDE. Eles apresentaram as bibliotecas GTK+ e o QT, respectivamente, que são a base para os gerenciadores de janelas mais conhecidos e utilizados por usuários de Linux. Depois do almoço, Marcos Pitanga, já conhecido por seu trabalho e livros, fez uma análise sobre *clusters* em Linux, explicando conceitos e dando exemplos práticos. O dia contou com inúmeras palestras e mini-cursos, mas falar de todos fugiria ao escopo deste resumo. Há uma descrição detalhada do evento (com várias fotos) em [1].

Ao final do evento, todas as organizadoras e integrantes do grupo LinuxChix-BR fecharam o programa com um debate sobre as realizações e os futuros rumos do projeto. Sulamita Garcia apresentou o histórico do grupo e as participantes responderam perguntas dos presentes sobre a própria motivação e existência do grupo.

Telefonia IP com Software Livre

No dia 30 de abril, a Livraria Tempo Real e a iSolve realizaram, no auditório da FIAP (Faculdade de Informática e Administração Paulista) em São Paulo, o primeiro evento abordando Telefonia IP com Software Livre. Mais de 300 participantes compareceram ao evento, que durou o dia todo.

A iSolve, em parceria com a Innovus, desenvolveu a programação técnica do evento, incluindo um workshop para demonstração prática das tecnologias de voz via Internet (VoIP – *Voice over IP*) que foram apresentadas nas palestras. A telefonia IP vem revolucionando a comunicação entre pessoas e empresas ao redor do mundo. A existência de diversos provedores de serviços de VoIP no mercado nacional e a tendência de forte crescimento para soluções nesse setor – estima-se que em 2005 os protocolos de voz irão corresponder a 40% do tráfego da Internet; em 2000, esse número era de apenas 5% – foram alguns dos motivos que despertaram o interesse dos participantes em comparecer ao evento. As apresentações,

realizadas pelas equipes das duas empresas, tiveram uma abordagem prática e contemplaram os ouvintes com uma visão geral da tecnologia por trás do conceito de VoIP, além de um workshop com instruções de instalação, configuração, recursos, customização, desenvolvimento e uso de soluções de voz via Internet baseadas em Software Livre – usando como ferramenta o PBX *Asterisk*, o mais maduro projeto de Software Livre para VoIP. Mais informações sobre o evento podem ser encontradas em [2].

XI Congresso de Informática Pública – CONIP

De 17 a 19 de maio foi realizado o XI Congresso de Informática Pública, o tradicional CONIP [3]. Abordando temas atuais como governança de TI, soluções de tecnologia móveis para o governo eletrônico (*m-Gov*) e cedendo um espaço privilegiado para o Software Livre, o CONIP se mostrou mais uma vez gerador de conteúdo e ao mesmo tempo fomentador de tendências de TI no âmbito governamental.

Com uma grade temática diversificada e a presença de diversas autoridades, o evento atraiu no ano passado mais de 2000 participantes. A expectativa este ano era de um número ainda maior (até o fechamento desta edição da Linux Magazine os números oficiais ainda não estavam disponíveis).

Alguns dos destaques do evento foram a palestra do atual CEO da Mandriva, François Bancelhon, que discorreu sobre o modelo de negócios da empresa e falou sobre suas fontes de receita. À mesa houve um debate interessante se desenvolveu entre o jornalista Luis Nassif, este editor que vos escreve e François sobre a viabilidade econômica e os modelos de negócios do Software Livre em geral.

Também foi destaque do evento a participação de Fernando Carmona, prefeito da cidade peruana de Miraflores, nos arredores da capital, Lima, e de Celso Junqueira, prefeito de Sud Menucci, cidade no noroeste do Estado de São Paulo. Ambos colocaram à disposição do cidadão uma estrutura *wireless* de alcance municipal, de maneira prática, eficiente e econômica. Celso Junqueira, em declaração à Linux Magazine, manifestou interesse em migrar algumas das tecnologias atualmente usadas na estrutura de TI da prefeitura para Software Livre. “Agora que está tudo funcionando, vamos procurar otimizar o que já temos. Aí é que entra o Software Livre”, afirmou, dizendo-se totalmente aberto à migração “onde fizer sentido”.

Uma outra grande novidade desse evento foi a assinatura do protocolo de intenções para disponibilizar o Sistema de Inventário CACIC (*Configurador Automático e Coletor de Informações Computacionais*), do Governo Federal, sob a GPL (Licença Pública Geral). O documento foi assinado pelo secretário de Logística e Tecnologia da Informação (SLTI) do Ministério do Planejamento, Rogério Santanna, e pelo presidente da Dataprev, Tito Cardoso de Oliveira Neto. O CACIC já foi instalado em 17 mil computadores da administração federal, mas seu compartilhamento estava restrito até agora aos órgãos do governo federal. A partir da assinatura do protocolo de intenções, o sistema poderá ser adotado também por órgãos públicos estaduais e municipais.

INFORMAÇÕES

[1] www.linuxchix.org.br/node.php?id=176

[2] eventos.temporeal.com.br/?area=5

[3] www.conip.com.br

[4] psl-mg.softwarelivre.org/emsl2005/programacao.php



Figura 2: Alguns dos presentes ao Encontro Mineiro de Software Livre 2005, realizado entre os dias 7 e 9 de abril em Belo Horizonte, Minas Gerais.

Encontro Mineiro de Software Livre

Nos dias 7, 8 e 9 de abril o campus da PUC Minas abrigou o Encontro Mineiro de Software Livre – EMSL 2005, com temário diversificado envolvendo governo e sociedade, inclusão digital, movimentos sociais, educação, saúde, desenvolvimento, ambiente operacional, corporativo e legislação.

O EMSL 2005 contou com quatro mesas temáticas, oito minicursos e mais de 25 palestras técnicas, referenciando o engajamento da comunidade e a participação de vários representantes governamentais. A programação do evento, com a grade de todas as palestras e minicursos ministrados, ainda se encontra disponível em [4].

Na manhã do primeiro dia, por ocasião da mesa temática *Ambiente de Trabalho Livre*, era esperada uma leve disputa entre os dois ambientes mais amigáveis utilizados no Software Livre: Gnome e KDE. A mesa se desenvolveu sem conflitos, predominando o clima de respeito mútuo e preocupação com o usuário. Houve um debate sobre questões de usabilidade dos diversos ambientes de trabalho de código aberto. Durante a tarde, em diversas palestras técnicas, foram abordadas questões envolvendo a migração para o Software Livre.

Nos outros dias, temas de fundamental importância para o desenvolvimento do Software Livre no Brasil foram tratados, entre eles: Software Livre em ambientes educacionais, o uso de Software Livre no sistema eleitoral, robótica, *clusters*, mineração de dados e, claro, Inclusão Digital.

CALENDÁRIO DE EVENTOS

EVENTO	DATA	LOCAL	WEBSITE
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 ENCOMP	01 a 05 de Agosto	Bonito, MS	www.enec.org.br
Wikimania 2005	04 a 08 de Agosto	Frankfurt, Alemanha	wikimania.wikimedia.org
1º BSD Day	13 de Agosto	São Paulo, SP	www.bsdday.com.br
Seminário: O impacto da escolha de tecnologia nos negócios corporativos	31 de Agosto	São Paulo, SP	www.linuxmagazine.com.br/seminario
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

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

III SOLISC

Data: 01 de Dezembro de 2005
Local: Florianópolis, SC
Website: www.softwarelivresc.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 3.500 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, e formatos vetoriais como SVG, AI e EPS para logotipos e ilustrações.

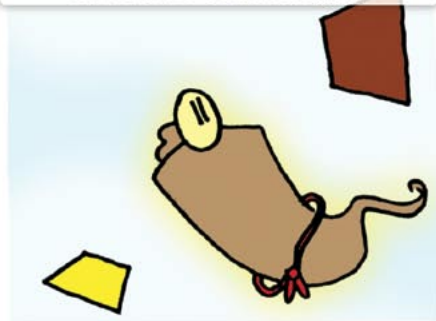
Uma revista passa por muitos estágios durante sua produção, e alguns meses podem se passar desde que seu artigo seja recebido até que a revista chegue às bancas. Portanto, nunca envie na última hora material ou notícias sobre encontros e eventos.

Envie suas colaborações para o endereço material@linuxmagazine.com.br. Evite enviar mais de 2 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

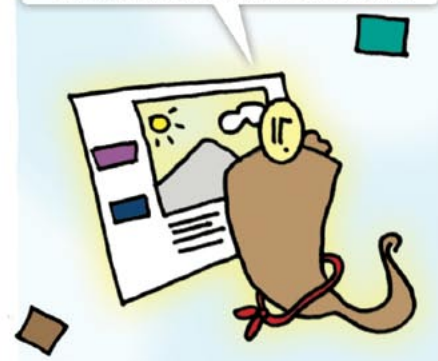
COMO TODOS SABEM, QUANDO MORREMOS
NOSSA VIDA É UPLODEADA PARA
UM BLOG NO CYBERESPAÇO...



HM... ENTÃO ESSE É O
BLOG DA MINHA VIDA?



EI, PERÁ! EU NUNCA ESTIVE NA BOLÍVIA.
E NUNCA NAMOREI NENHUMA CACÁ...



Ó, TORVALDS,
TODO-PODEROSO!



O QUE QUE FOI, Ô
RECÉM-FALECIDO?



MINHA VIDA FOI
PUBLICADA
NUM WIKI!



ÍNDICE DE ANUNCIANTES

SERVIÇOS

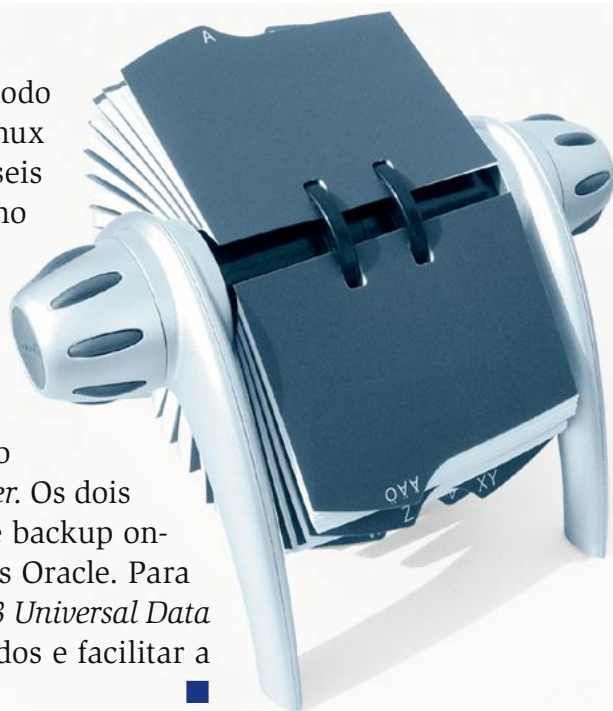
ÍNDICE DE ANUNCIANTES		
ANUNCIANTE	SITE	PÁGINA
A Casa do Linux	www.casadolinux.com.br	96
Alternativa Linux	www.alternativainux.com.br	61
Conisli	www.conisli.org.br	09
Dell	www.dell.com.br	100 (quarta-cap)
Ermida	www.ermida.com.br	96
Green Treinamento e Consultoria	www.green.com.br	19
Infomedia	www.infomediatv.com.br	77
Linux Corp	www.rpmbrazil.com.br	99
Linux New Media	www.linuxnewmedia.com.br	02 (contra-cap), 51, 55
Linux World	www.linuxworldexpo.com	99 (terceira-cap)
Unicial / LinuxPress	www.unicial.com.br	73

Agosto de 2005: décima primeira edição

Na próxima edição...

Bancos de Dados

Bancos de Dados ocupam o centro das atenções em quase todo departamento de TI que se preze. A edição de agosto da Linux Magazine vai dedicar a mesma atenção a esse tema: são seis artigos abordando com riqueza de detalhes questões como desempenho, segurança, estabilidade, backup e migração de dados *de e para* as diferentes plataformas e sistemas de bancos de dados. Um artigo sobre o recém-lançado PostgreSQL 8 abre as matérias de capa da edição 11. Um artigo sobre *clusters* de bancos de dados, desta vez usando MySQL, continua a nossa viagem por esse terreno, seguido de perto por uma matéria sobre o *MySQL Backups Manager*. Os dois artigos seguintes tratam, respectivamente, de técnicas de backup online e dão dicas para a “sintonia fina” de bancos de dados Oracle. Para encerrar, apresentamos o conversor de “dialetos” SQL *Eva/3 Universal Data Converter*, que promete pôr fim à Babel dos bancos de dados e facilitar a migração de um sistema para outro. ■



Honeypot, o “Boi de Piranha” da sua rede

Um *honeypot* é um sistema que possui falhas de segurança reais ou virtuais, colocadas de maneira proposital de modo a permitir uma invasão. Com isso obtém-se resultados interessantes em situações reais e pode-se traçar um perfil dos ataques e do comportamento dos atacantes. Neste artigo, mostraremos como montar um honeypot usando Linux. ■

OpenWRT e roteadores WLAN

Os *firmwares* originais oferecidos pelos fabricantes, ajustados individualmente a cada roteador, possuem um leque de funções muito limitado e são extremamente direcionados a um conjunto particular de objetivos. Quando o firmware padrão do roteador não consegue fazer tudo o que você quer, é possível substituí-lo por uma distribuição do Linux mais adequada às suas necessidades. O *OpenWRT* é desenvolvido sob a GPL e constitui mais do que somente um firmware para roteadores com Linux embarcado: trata-se de uma mini-distribuição completa, inclusive com gerenciamento de pacotes, que pode ser ampliada quase que *à la carte*.

De maneira quase imperceptível para o grande público, cada vez mais fabricantes utilizam roteadores WLAN baseados em Linux. Nosso artigo mostra que há uma variedade considerável de roteadores WLAN e DSL conhecidos no mercado que rodam Linux e para os quais há firmwares adequados. ■

Macromedia ColdFusion MX 7

O *ColdFusion MX 7 Server* da Macromedia pode criar aplicativos web de alta disponibilidade e performance. Originalmente um aplicativo Windows® escrito em C++, o programa passou a ser desenvolvido com base no J2EE e roda também no Linux, no AIX, no Mac OS X e no Solaris. Vamos dar uma olhada em sua versão para Linux em nossa próxima edição. ■



CD Número 10
julho 2005

LINUX

MAGAZINE

MediaWiki, Zope, Plone, Drupal, TWiki

Incluindo todos os plugins e add-ons necessários
para você construir um ambiente completo de
gerenciamento de conteúdo.



COMPACT
disc

PRODUZIDO
NO POLO INDUSTRIAL
DE MANAUS
CONHEÇA A AMAZÔNIA



Produzido no Polo Industrial de Manaus por Sonopress Rimo da Amazônia Ind. e Com. Fonográfica LTDA - Indústria Brasileira - CNPJ: 84.494.129/0001-93 - sob encomenda de Linux New Media LTDA - 06.351.943/0001-45

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 nº10 e não pode ser vendido separadamente.