

LINUX MAGAZINE

NÚMERO 2

Lua
Slackware 10
PlanMaker
Kurumin
Debian
LPRng
BitTorrent
Sistemas de Arquivos

O fim das limitações, perca o medo do *fsck*

SISTEMAS DE ARQUIVOS

Ext 3, ReiserFS, XFS, JFS

Testes comparativos de desempenho

7x mais rápido: turbine seu HD

Ferramentas indispensáveis

Criptografia de bolso

BitTorrent

Acelere a distribuição de seu software sem gastar mais com isso

LPRng

Aprenda a configurar este servidor de impressão

Veja também:

Crie um repositório Debian

Personalize o Kurumin

Testamos o PlanMaker

Analizamos o Slackware 10

Lua: a linguagem de outro mundo



Linux Magazine R\$12,90
 Edição Número 2 Setembro 2004
 00002
 9 771806 942009

Excelência



Prezado leitor, prezada leitora da Linux Magazine,

Recentemente, em diálogos esparsos entre alguns membros da redação da Linux Magazine, bem como em conversas com representantes da

indústria e do mercado de TI e com a comunidade, o tema da qualidade do software de código aberto e livre tornou-se uma constante.

Em um momento em que o modelo de exploração comercial de software mostra visíveis sinais de cansaço, a maioria maciça das grandes empresas que atuam na área de tecnologia de informação, seja oferecendo produtos, seja prestando serviços, estão adotando ou desenvolvendo, se não no todo, pelo menos em parte, sistemas baseados em software de código aberto, no intuito de reduzir custos de produção e desenvolvimento, bem como de aumentar a segurança, a qualidade, a compatibilidade e a auditabilidade de seus produtos. Alcatel, AMD, AOL, Apple, Borland, Cisco Systems, Computer Associates, Corel, Cyclades, Dell, EMC, Ericsson, Fujitsu, Google, Hewlett-Packard, Hitachi, IBM, Intel, Mitsubishi Electric, Motorola, NASA, NEC Corporation, Nokia, Novell, NTT, Oracle, Real Networks, Samsung, SAP, Sharp, Siemens, Silicon Graphics, Sony, Sun Microsystems, Toshiba, Transmeta, VIA Technologies, Yahoo, ... a lista se estende quase que indefinidamente.

Questões sobre a viabilidade do modelo econômico em torno do software livre, levantadas com frequência no passado, tornaram-se mais raras devido à maturidade dos sistemas de código aberto. O Linux desponta como exemplo mais notório das possibilidades produzidas por este modelo. Disponível para (no mínimo) quinze arquiteturas e presente em equipamentos que vão desde PDAs até mainframes, o espaço conquistado pelo pingüim aumenta a cada dia.

Mas existem outros exemplos, como o OpenOffice.org, o pacote de programas para escritório multi-plataforma de código aberto, que conquista cada vez mais espaço em empresas, instituições governamentais e em máquinas de usuários domésticos. Temos também o navegador Mozilla, o servidor web Apache (o mais usado no mundo), diversos servidores de e-mail (Sendmail, Postfix, Exim, Qmail, ...), dúzias (literalmente) de ambientes desktop, sistemas ERP e CRM (tais como Compiere, SQL-Ledger e GNU Enterprise), além de várias soluções groupware (OpenGroupware, Kolab, OPEN-XCHANGE, phpGroupware, entre outros), gerenciadores de bancos de dados relacionais (PostgreSQL, MySQL, Firebird, Ingres, MaxDB - ex-SAP DB, HSQLDB, Cloudscape) e centenas de outros aplicativos.

Recentemente, o Gartner Group previu que, a partir de 2007, as empresas desenvolvedoras de software para UNIX encerrarão paulatinamente o suporte aos UNIX tradicionais e passarão a oferecê-lo somente para Linux.

Com isso, a polarização entre o software de código aberto e o software proprietário de código fechado vai se acirrar. A grande batalha que já atingiu o mercado de servidores vai invadir o desktop corporativo e acabar na casa do usuário doméstico. Os grandes projetos de código aberto e livre já atingiram níveis de excelência excepcionais, que atendem perfeitamente a todas as necessidades na empresa ou na sua casa. Só falta a malha de serviços e produtos em torno deles atingir o mesmo nível de excelência.

No âmbito corporativo, a implementação de soluções baseadas em software de código aberto não pode prescindir da metodologia e profissionalidade durante o processo de desenvolvimento do projeto, da mesma forma que ocorre quando grandes corporações implementam soluções baseadas em sistemas proprietários. O problema hoje não está no software!



Nós nos orgulhamos das origens de nossa publicação: ela remonta aos primeiros dias daquilo que ficou conhecido como a Revolução Linux.

A versão em alemão da Linux Magazine, fundada em 1994, foi a primeira revista dedicada exclusivamente ao Linux na Europa. Desde então nossa experiência e conhecimento, bem como nossa rede de contatos dentro da comunidade Linux, cresceram e se expandiram pelo mundo todo.

Como leitor da Linux Magazine, você está se integrando a uma rede de informações dedicada a distribuir conhecimento e experiência técnica em torno do Linux e do Software Livre. Entretanto, nós não estamos apenas dizendo o que acontece nesta área: nós fazemos acontecer!

Para isso, precisamos, como em qualquer outra área, de profissionais capacitados, com bom domínio da tecnologia que utilizam. Abrir mão disto é prestar um desserviço ao software de código aberto, pois, no final, a percepção do usuário é sempre de que o novo sistema, que é "diferente" ou "exótico", apresenta problemas. Infelizmente, tivemos vários exemplos disto no passado.

A tendência é clara: com raríssimas exceções, a indústria e o mercado estão abraçando o software livre. A excelência tecnológica está aí. A excelência nos serviços e no suporte está chegando. Quem reconhecer a oportunidade e investir em uma política de treinamento consistente, vai se diferenciar da concorrência e, a custos mais baixos, conquistará seu espaço no mercado.

Quando chegarmos a este nível, até os grandes ausentes da lista de empresas citadas terão que entrar no trem da modernidade... ou ficar na estação, na esperança de um dia um outro passar.

Rafael Peregrino da Silva
Editor

NOTÍCIAS

Gerais	6
Negócios	12
Adoção total do Linux: Entrevista com o Vice-Presidente de tecnologia da Oracle	
(In)Segurança	14
Kernel	16

CAPA

Bloco da alegria	19
Provão!	23
Nossa extensa análise mostra as qualidades e defeitos de cada um dos principais sistemas de arquivos para o Linux.	
Oficina de formatação	26
Cada sistema de arquivos tem suas próprias ferramentas para manutenção e ajustes. Saiba como tirar proveito delas.	
Top secret	31
Torne seus dados indecifráveis com o uso de técnicas de criptografia e um simples chaveirinho USB.	
Pé na tábua	36
Aprenda a usar, de forma segura, o <i>hdparm</i> para melhorar o desempenho dos seus discos rígidos IDE.	

Jogue seus dados	38
Tenha espaço em disco "ilimitado" com a dupla LVM e RAID.	

ANÁLISES

Slack 10: a bola da vez	42
PlanMaker: a arte das planilhas	46
Uma alternativa comercial ao OpenOffice Calc.	

TUTORIAL

Embrulha pra presente!	48
LPRng - A nova geração	50
Remasterizando o Kurumin	56
BitTorrent: carga distribuída	62

18 Um lar para seus arquivos

Os modernos sistemas de arquivos para o Linux dispõem de um recurso chamado "journaling", um registro de todas as alterações efetuadas em seus dados. Com ele, o risco de perda de informações devido a um desligamento inadequado é reduzido e verificações de integridade ocorrem de forma muito mais rápida. Em nossos artigos de capa você saberá mais sobre as características desses sistemas, como melhorar seu desempenho, manter seus dados seguros e aumentar sua capacidade de armazenamento de forma quase ilimitada.



42 A bola da vez

Com a última palavra em software, a nova versão do Slackware, a distribuição Linux mais antiga ainda em atividade, prova que ainda tem fôlego de sobra para brigar com as grandes distros comerciais. Nossa análise detalhada, feita em oito máquinas, conta tudo o que acontece durante o processo de



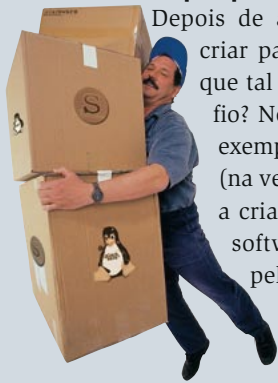
instalação, mostra as principais novidades nas interfaces gráficas, navegação e multimídia e o ajuda a se livrar de "armadilhas" e "pegadinhas".

84 É hora do show!

Uma imagem vale mais do que mil palavras, principalmente se ela vai ser usada como auxílio no treinamento de usuários ou para demonstrar os recursos da nova versão de um programa. Mas que tal substituir os tradicionais screenshots por algo mais animado, como vídeos? Para isso temos o XVIDCap, capaz de capturar vídeos nos formatos MPEG1, MPEG2 e MPEG4 de regiões do seu desktop, que depois podem ser gravados em CD ou colocados para download na Internet.



48 Embrulha pra presente!



Depois de aprender, na edição passada, a criar pacotes básicos para o Slackware, que tal aumentar um pouquinho o desafio? Nesta edição vamos aprender, com exemplos, a criar *slackbuilds*, arquivos (na verdade, Shell Scripts) que facilitam a criação e manutenção de pacotes de software no formato “oficial” usado pelo Slackware. Que tal usar esse conhecimento para “empacotar” um software que não faz parte da distribuição e beneficiar toda a comunidade?

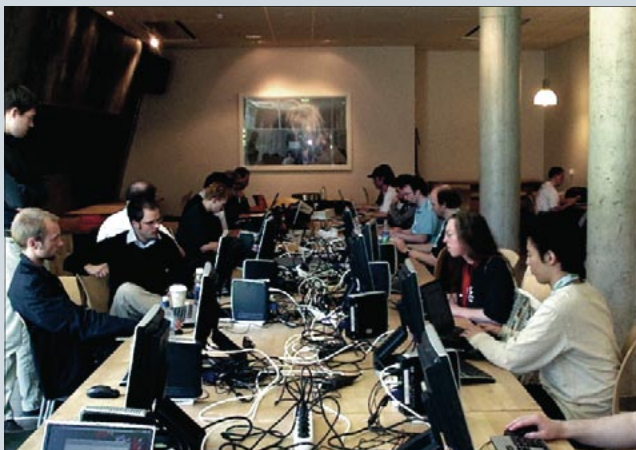
56 Pinguim ao gosto do freguês

Imenso sucesso de público, graças ao pequeno tamanho e capacidade de rodar direto de um CD, O Kurumin, criação do brasileiro Carlos Morimoto, herdou de seu “pai”, o Knoppix, uma característica que lhe dá imensa flexibilidade: a facilidade de “customização” para os mais variados fins. De uma simples mudança da página inicial do navegador à instalação e remoção de software, tudo é possível. Veja como na primeira parte desta série de artigos.



93 GUADEC 2004

Saiba tudo o que rolou durante a GUADEC 2004, a conferência européia de usuários e desenvolvedores do GNOME que aconteceu em Kristiansand, na Noruega. Telsa Gwinne, uma das principais “beta testers” deste ambiente desktop, esteve lá e nos conta sobre as discussões, os projetos e, claro, as festas que aconteceram nos três dias do evento.



Paulino Michelazzo, cortesia

SYSADMIN

- Coluna do Charly** 64
O PhpLDAPAdmin facilita o gerenciamento de diretórios LDAP.
- Formando Pares**..... 66
- Espelho, espelho meu...**..... 68
Monte seu próprio repositório de pacotes Debian.

PROGRAMAÇÃO

- Feitiço da Lua**..... 71
Uma linguagem de programação nacional usada para automação, em dispositivos embarcados e até grandes jogos.

LINUX USER

- Xandros Desktop** 78
Em nosso CD, uma das distribuições Linux mais fáceis de usar.
- Gráficos em modo texto**..... 80
Acelere a conversão de imagens com o ImageMagick.
- Seleção VI-sual**..... 82
Saiba mais sobre esse poderoso modo de operação do VI.
- É hora do show!**..... 84
Saudades do Lotus ScreenCam? O XVIDCap dá conta do recado.

- Papo de Botequim**..... 87
Tudo sobre *grep*, e vamos montar nosso primeiro aplicativo.

COMUNIDADE

- Unindo esforços** 92
A importância dos grupos de usuários na comunidade do Software Livre.
- Melhor do que nunca** 93
Saiba o que rolou durante a GUADEC 2004.
- O pingüim e o rio** 94
Como uma pequena cidade informatizou a administração pública e diminuiu a exclusão digital com Software Livre.

SERVIÇOS

- Eventos / Índice de Anunciantes** 96
- Expediente Editorial / Quadrinhos**..... 97
- Na Próxima Edição...**..... 98

Mundo Livre em Revista

■ Tribunal valida a GPL

Em uma batalha jurídica contra a fabricante de roteadores Sitecom, o projeto de código aberto netfilter/iptables alcançou uma grande vitória: com o veredito de 19 de maio de 2004, o tribunal da comarca de Munique confirmou um recurso pelo qual a Sitecom não pode mais comercializar, na Alemanha, seus roteadores WLAN que fazem uso do netfilter/iptables, mas que não respeitam os termos e condições da GNU General Public License, a GPL. O valor das custas do processo também foi confirmado pelo veredito: 100.000 Euros.

Mas a grande vitória alcançada por esse veredito foi o reconhecimento da validade jurídica da GPL na Alemanha. Nas palavras do próprio tribunal: “A câmara concorda com a percepção do requerente de que as condições exaradas na GPL não significam de modo algum renúncia de direitos autorais e de propriedade intelectual. Portanto, é legítima a ação do processante para fazer valer seus direitos autorais”.

Ainda não se sabe se a fabricante de roteadores irá recorrer da decisão. ■

http://www.jbb.de/judgment_dc_munich_gpl.pdf

■ Cherry lança teclado especial para Linux

A empresa fabricante de teclados Cherry trabalha atualmente no projeto de um teclado orientado às necessidades de usuários Linux. Para tanto, fechou um acordo de cooperação com a SuSE, maior distribuição Linux da Europa. O novo teclado, o modelo *Cherry CyMotion Master Linux*, deverá entrar no mercado alemão já a partir do início de setembro e na Irlanda e Reino Unido até o fim deste ano. De acordo com a empresa, o teclado oferecerá finalmente a usuários de Linux o mesmo conforto de digitação de que desfrutam há anos os usuários de sistemas Microsoft. O que isso significa na prática são 29 teclas especiais programáveis para qualquer função e que podem ser usadas, por exemplo, para ativar programas multimídia e chamar funções utilizadas mais frequentemente em aplicações de escritório ou no navegador de Internet, bem como para executar programas ou mesmo utilizar áreas de trabalho diferentes no desktop do usuário.

As teclas especiais para multimídia e Internet ficam na parte superior do teclado. As de multimídia servem para controlar a reprodução de músicas e filmes por diferentes media players (fun-



ções para iniciar, parar, voltar ou avançar) e as de Internet servem para chamar máquinas de busca ou sites pré-definidos, bem como para pular para frente e para trás durante a navegação. Na parte inferior do teclado encontram-se as teclas para controle de volume e “mudo” e para a escolha do *media player*.

Teclas chamadas “Xpress Keys” podem ser programadas para chamar funções como cortar, copiar, colar, desfazer e refazer. Além disso há uma tecla para ejetar o CD, tal como no teclado do Macintosh, uma tecla @ em separado e, no lugar onde costuma ficar a tecla com o símbolo do Windows, temos a simpática figura do Tux, que deverá servir para chamar o menu de desktops como GNOME ou KDE.

O preço na Europa será de 39,95 Euros e o pacote inclui, além do teclado, o programa *KeyM@n* para a programação das teclas e a distribuição SuSE Linux 9.1 Special Edition. ■

■ Cloudscape é liberado sob licença da ASF

Durante a LinuxWorld Conference & Expo, que ocorreu nos Estados Unidos em São Francisco entre os dias 2 e 5 de agosto deste ano, a IBM abriu o código do seu sistema de gerenciamento de banco de dados relacional (RDBMS) Cloudscape, deixando a sua manutenção nas mãos da Apache Software Foundation (ASF). A ASF vai colocar o projeto em sua incubadora de projetos sob o codinome Derby e verificar se as 500.000 linhas de código do sistema são compatíveis com a sua licença.

O Cloudscape foi adquirido pela IBM em 2001 por ocasião da compra da Informix. Ele é utilizado nos produtos Workplace e na família Websphere (portal e servidor de aplicações), é totalmente desenvolvido em Java, e não oferece concorrência ao DB2, apesar de contar todos os recursos de um banco de dados relacionais - transações, em especial.

Ocupando apenas 2 MBytes de espaço em disco, o Derby é especialmente adequado para sistemas embarcados, mas também pode ser usado para sistemas de desenvolvimento e testes, já que o código desenvolvido para ele é compatível com o DB2 - e vice-versa. Por essa razão, todas as aplicações para DB2 funcionam nele também. A IBM espera que a comunidade contribua para o desenvolvimento posterior do Derby e que ele se torne um dos líderes entre os bancos de dados de código aberto. A empresa continuará a oferecer versões comerciais do Derby sob o nome Cloudscape, que terá como base a versão livre, mas disporá de mais recursos. ■

<http://www-306.ibm.com/software/data/cloudscape/>

<http://incubator.apache.org/derby/>

■ RealPlayer 10 para Linux

Tanto o RealPlayer 10 quanto o Helix Player 1.0, projeto de código aberto que serve de base para a versão para Linux do media player da Real Networks, já podem ser baixados dos respectivos web

sites da empresa. A Novell/SuSE e a Red Hat já informaram estar dispostas a incluí-los nas próximas versões de suas distribuições Linux.

O RealPlayer 10 oferece suporte para os formatos RealAudio, RealVideo, MP3, Ogg Vorbis, Ogg Theora, H.263 e AAC. Um plug-in para o Mozilla possibilita a integração do media player dentro do navegador sem que haja necessidade de uma janela extra para tocar a mídia. O Helix Player não suporta MP3 nem os formatos proprietários da Real Networks. Versões experimentais dos dois players para Solaris (Sparc) e Linux PowerPC também estão disponíveis. ■

<http://www.real.com/linux/>

<https://player.helixcommunity.org/>

■ Unidas migra para Linux e Java

A Unidas Aluguel de Carros, uma das maiores empresas de locação de veículos do Brasil, decidiu apostar na utilização de software livre e aplicações para ambiente Web para reduzir os custos e modernizar seus sistemas. Duas das aplicações desenvolvidas em Java, a Central de Reservas e o Portal de Relacionamento, já estão implementadas.

Segundo Marcelo Testa, diretor de Tecnologia da Unidas, a decisão de migrar os servidores da empresa para Linux e implementar soluções Web desenvolvidas em Java tinha como objetivo a adoção de um ambiente multiplataforma e evitar a formação de um vínculo muito estreito com apenas um fornecedor (o chamado *lock-in*). Até o final de 2004, todos os 400 desktops da empresa também deverão estar rodando Linux.

A empresa investiu aproximadamente R\$ 400 mil reais no projeto, sem contar a aquisição de hardware. Em contrapartida, apenas com licenças a Unidas economizará anualmente algo entre R\$ 300 mil e R\$ 400 mil.

A ferramenta RAD multiplataforma GeneXus foi a escolhida para o desenvolvimento das soluções de que a empresa necessitava. Ainda segundo Testa, ela apresenta um custo baixo, pois o investimento na aquisição da ferramenta é amortizado pelo tempo de desenvolvimento e pelo menor número de pessoas trabalhando no código. ■

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

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



■ Bombril migra para Linux

No final de 2003 havia na Bombril a necessidade de atualização do sistema de correio eletrônico da empresa (que usava o MSMail) para uma nova tecnologia. A solução para atender a esta necessidade foi migrar o servidor de correio eletrônico para Qmail, rodando na plataforma Linux.

Com a adoção da plataforma Linux foi possível também atender a outras demandas dentro da Bombril e ferramentas novas foram introduzidas, tais como acesso remoto via WebMail, Aviso de Ausência (férias) e lista de endereços central via OpenLdap. Além disso, a infraestrutura da rede foi otimizada, eliminando-se os gateways até então necessários para a troca de emails com as unidades de Recife e Minas, o que melhorou a performance de troca de mensagens internas e entre as filiais. A segurança interna em relação a possíveis contaminações por vírus foi aumentada através da adoção de análise de vírus em todos os emails internos, filtro de anexos e de SPAM.

“Conseguimos migrar gradativamente todos os usuários do MS MAIL para o ambiente Linux sem criar nenhum impacto na organização, trazendo melhorias significativas na solução técnica, assim como nos recursos disponíveis para os usuários finais.”, relata Marcelo de Cillo, CIO da Bombril.

O projeto de migração da empresa foi desenvolvido pela empresa 4Linux, segundo a qual hoje existem novas necessidades dentro da Bombril onde o uso do Linux e de outros softwares livre estão sendo considerados. ■

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

<http://www.4linux.com.br/>

■ Lançada versão 3.3 do KDE

A versão 3.3 do ambiente de trabalho KDE já está disponível para download. O código-fonte pode ser baixado do site oficial do projeto, e várias distribuições Linux já lançaram pacotes binários. De acordo com as informações da equipe de desenvolvimento, a versão 3.3 traz diversas melhorias na usabilidade, estabilidade e integração entre os vários aplicativos do desktop. Nos últimos seis meses, mais de 7.000 bugs foram corrigidos e mais de 2.000 novos recursos foram implementados.

O cliente de e-mail Kmail, finalmente, permite escrever mensagens em HTML e seu sistema de criptografia de mensagens foi completamente reescrito, oferecendo recursos que facilitam a codificação, decodificação, assinatura e verificação de assinaturas em e-mails utilizando ferramentas e protocolos comuns, tais como PGP, GPG (Gnu Privacy Guard) e S/MIME. O suporte ao protocolo IMAP está maduro e os controles de acesso estão integrados ao cliente.

A principal novidade no KDE 3.3 está em suas ferramentas de gerenciamento de informações pessoais (PIM): o KDE dispõe agora de uma solução groupware completa e integrada. Todas as ferramentas de comunicação pessoal, tais como KMail, KOrganizer, KAddressBook e KNode são componentes de um sistema único e criam um ambiente integrado, chamado Kontact, similar ao Microsoft Outlook ou ao Lotus Notes. A integração com o servidor Kolab permite que calendários e contatos sejam arquivados no servidor e compartilhados com outros usuários. Reuniões podem ser marcadas e automaticamente colocadas nos calendários dos participantes e informações sobre conflitos entre horários podem ser acessadas facilmente.

O K Desktop Environment está disponível em 50 idiomas, com traduções para mais 30 já parcialmente prontas. A versão 3.3 ficou pronta a tempo para o lançamento durante o KDE Community World Summit 2004, evento que ocorreu de 21 a 20 de agosto em Ludwigsburg, na Alemanha, e reuniu desenvolvedores KDE do mundo inteiro. ■

<http://www.kde.org/announcements/>

[announce-3.3.php](http://www.kde.org/announcements/announce-3.3.php)

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

<http://conference2004.kde.org/>

■ ReiserFS v4 otimiza a manipulação de arquivos

A versão 4 do sistema de arquivos ReiserFS já está disponível para download na página do projeto. Lá você encontrará também informações sobre como integrá-lo ao kernel 2.6. Segundo o site do projeto, a nova versão ainda não deve ser usada em ambientes de produção.

Segundo os desenvolvedores do sistema, o Reiser4, como a nova versão é chamada, seria o sistema de arquivos mais rápido existente para Linux. Esta afirmação é documentada com alguns testes (“benchmarks”) simples. De acordo com esses benchmarks, principalmente em operações simultâneas de leitura e escrita de várias “streams” de dados, o Reiser4 se mostra superior ao ext2, ext3 e à sua versão anterior (v3): segundo o benchmark “slow”, o sistema tem desempenho similar aos seus concorrentes quando apenas uma stream está sendo lida ou escrita. No caso, por exemplo, de oito streams simultâneas, o Reiser4 é até sete vezes mais rápido na leitura e até quatro vezes mais rápido na escrita de dados.

Além disso, o Reiser4 é um sistema de arquivos atômico, o que significa que operações no sistema de arquivos ou ocorrem inteiramente ou simplesmente não ocorrem, de modo que o sistema não pode ser corrompido por operações que ocorram pela metade. Isso acontece sem perdas significativas de desempenho devido ao uso de algoritmos otimizados, que não copiam os dados duas vezes durante as operações.

O armazenamento de metadados não é mais realizado com as conhecidas árvores B*, da versão 3, mas com as “Dancing Trees” (árvores dançantes), cuja utilização deve acelerar o acesso a arquivos e diretórios. Adicionalmente, para conseguir uma escalabilidade maior, arquivos pequenos não preenchem mais blocos inteiros, mas podem ser concatenados em um só bloco, o que evita o desperdício de espaço em disco.

A nova versão também oferece uma interface para extensões que permite expandir o sistema com novas funções. Segundo a equipe de desenvolvimento, o código passou por uma extensa fase de controle de qualidade e todas as suas funções devem ser seguras e simples de auditar – a chamada “military-grade se-

curity”, ou segurança em escala militar. Por coincidência, um órgão de pesquisa do exército norte-americano, a DARPA, patrocina o projeto. ■

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

<http://www.jburgess.uklinux.net/slow.c>

■ OpenOffice 2.0 terá banco de dados próprio

Demorou, mas está decidido: o HSQLDB será o banco de dados que acompanhará a versão 2.0 do mais popular pacote para escritório de código aberto.

Há muito tempo que o OpenOffice.org já serve de interface gráfica para bancos de dados. O problema é que tais bancos de dados precisam estar disponíveis no sistema. Até agora o projeto não tinha como criar um banco de dados próprio, como faz o Microsoft Access. Isso acabará a partir da versão 2.0, com a qual será possível criar bancos de dados pessoais na sua máquina local. A vantagem do Microsoft Professional Office, com o popular Access, e a do StarOffice, com o Adabas DB (proprietário), será em breve coisas do passado.

O HSQLDB é o banco de dados relacional SQL escrito em Java mais largamente utilizado. Ele possui um driver JDBC e fornece suporte para um grande subconjunto de comandos SQL ANSI-92 (formato BNF em árvore), SQL 99 e 2003. Oferece um banco de dados pequeno (menor que 100 KBytes) e rápido, cujas tabelas podem ser escritas em disco ou mantidas em RAM. Adicionalmente, dispõe de um servidor web simples e de ferramentas de acesso à base de dados em RAM, bem como de gerenciadores do banco de dados, que podem ser executados como applets. Sua versão atual é a 1.7.2, e seus termos de licenciamento são baseados na licença BSD.

Outras novidades da versão 2.0 do OpenOffice: conexões com bancos de dados externos foram melhoradas e simplificadas, acessos via SQL foram refinados. Um formulário pode ser salvo separadamente, sem que uma aplicação do OpenOffice tenha de ser especialmente aberta para este fim. Com isso pode-se criar excelentes interfaces e ferramentas de administração para bancos de dados já existentes (como por exemplo para um banco de dados MySQL ou PostgreSQL). Além disso, a nova versão também oferece suporte para conexões

avançadas com servidores LDAP - não é mais necessário utilizar o usuário *anonymous*: pode-se usar autenticação normalmente e, para completar, o acesso pode ser feito via SSL. ■

<http://hsqldb.sourceforge.net/>

http://www.amt-wiesbaden.de/downloads/OOo_20_Vorschau.pdf

■ Nações Unidas apóiam Software Livre através de nova agência

A International Open Source Network (IOSN), uma iniciativa da United Nations Development Programme (UNDP), acaba de lançar um guia para iniciantes em Linux. O documento, em inglês, se dirige especialmente a futuros usuários de Linux de países em desenvolvimento e foi publicado sob a licença Creative Commons Attribution License, com o intuito de estimular a sua popularização e traduções para outros idiomas.

O guia lançado não se dirige apenas a iniciantes de Linux no desktop, mas também àqueles que ainda não possuem nenhuma experiência com um PC. Os usuários devem, entre outras coisas, aprender a gerenciar arquivos, a trabalhar com o OpenOffice e como navegar na Internet utilizando o Linux. Além disso, os usuários devem ser estimulados a continuar a aprender sozinhos após terem encerrado o aprendizado inicial com o guia, que tem 13 capítulos e um anexo sobre KDE, e está disponível em formato PDF e como documento do OpenOffice. Para instrutores há também um documento em forma de apresentação.

A IOSN é um centro de competência para software de código aberto e livre na região asiática em torno do Pacífico, que tem como objetivo principal promover o uso desse tipo de software em países em desenvolvimento. Segundo a nova agência, o “software livre e aberto é uma solução que possibilita inclusão digital a países em desenvolvimento”, “representando uma oportunidade para esses países adotarem softwares e soluções acessíveis e diminuir o vácuo digital entre eles e os países desenvolvidos. O software livre é uma alternativa à importação de software e de aplicativos proprietários caros.” ■

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

<http://www.iosn.net/training/end-user-manual/>

■ Pepper Pad 2: Um Tablet PC com Linux

A empresa Pepper Computer Inc. acaba de lançar um Tablet PC rodando na plataforma Linux: o Pepper Pad 2 é um computador portátil sem fio, desenvolvido para o uso doméstico ou em trânsito. O MontaVista® Linux® e o ambiente Java são a base para o sistema operacio-

nal “invisível”, segundo a terminologia utilizada pela empresa, que foram otimizados para o processador Intel® Xscale® de 624 MHz do Tablet PC. Ele dispõe de uma tela TFT SVGA de 8,4 polegadas sensível ao toque, com resolução de 800 x 600 pixels, de modo que páginas na Internet podem ser mostradas inteiramente e filmes podem ser assistidos sem

dificuldades. Há também um teclado QWERTY, que foi dividido ao meio, com cada uma de suas partes colocada de um lado da tela.

O Pepper Pad 2 pesa menos de 1 kg, dispõe de um disco rígido de 20 GBytes, 256 MBytes de SDRAM, 32 MBytes de memória de vídeo e 32 MBytes de memória flash, de modo que ele pode armazenar várias horas de música e vídeo, além de fotos e páginas da Internet. Ele dispõe de uma bateria recarregável de polímero de lítio com gerenciamento de energia, uma fonte de alimentação AC, suporte a Bluetooth, infravermelho e USB 1.1 embutidos, além de permitir acesso à redes sem fio no padrão WiFi 802.11b + g. Além disso, ele vem com sistema multimídia estéreo e microfones embutidos, bem como saída para fone de ouvido, entrada para microfone externo e um gamepad de 5 botões. O gabinete inteiro é reforçado e à prova de líquidos para resistir aos rigores do uso diário. O preço do brinquedo é 800 dólares nos Estados Unidos. ■

Pepper Pad:

http://www.pepper.com/pepper_pad/



■ Lançado o FreeNX

Foi lançada durante a aKademy 2004 uma versão livre do NX, servidor de terminais que permite o rápido acesso à interface gráfica de aplicações X em tela cheia, mesmo através de linhas de baixa velocidade (9600 bps), tais como conexões discadas ou via GSM.

Apesar da firma italiana NoMachine comercializar licenças do servidor NX para Linux e Solaris, oferecendo gratuitamente apenas o cliente para a tecnologia para diversas plataformas, ela liberou os algoritmos e técnicas que servem de base para a implementação do servidor sob a GPL. Essas técnicas consistem na aceleração do protocolo X e de diversos outros protocolos (RDP, ICA e VNC) através do que se convencionou chamar de “proxy caches” e “proxy agents” e na transferência superotimizada de dados entre servidor e cliente. O FreeNX é a primeira implementação livre baseada no código liberado sob a GPL. ■

<http://kalyxo.freedesktop.org/twiki/bin/view/Main/FreeNX>

<http://www.nomachine.com/documentation/intr-technology.php>

■ IBM põe em operação um dos maiores mainframes Linux da Europa

A empresa Endress + Hauser, especializada em sistemas de medição e automação em tecnologia de processos industriais, está migrando suas aplicações SAP, que até então rodavam em sistemas convencionais UNIX, para dois mainframes IBM zSeries (z990), rodando Linux. Para a IBM isso é um claro sinal da indústria de que o Linux já é uma tecnologia madura em equipamentos mainframe para sistemas de missão crítica.

Ambos os sistemas, que contam com 328 GByte de RAM, serão utilizados por 3.500 usuários. As 19 aplicações SAP estão distribuídas em 14 partições lógicas (LPARs) e os bancos de dados DB2 utilizados por elas ocupam outras 6 LPARs. Os servidores de aplicações usam 36 processadores da linha zSeries rodando Linux (IFL, Integrated Facility for Linux). De acordo com a IBM, esta é uma das maiores instalações de mainframe Linux existente na Europa.

A arquitetura especial da linha zSeries permite a virtualização completa de hardware em sistemas operacionais

z/VM, o que possibilita a operação paralela de centenas de sistemas Linux. Além disso, os sistemas podem ser expandidos com hardware adicional sem interrupção de funcionamento e podem ter seu desempenho aumentado ou reduzido através da ativação ou desativação de seus componentes, respectivamente. ■

<http://www.de.endress.com/>

<http://www-1.ibm.com/servers/eserver/zseries/>

<http://www-1.ibm.com/servers/eserver/zseries/os/linux/>

<http://www-1.ibm.com/servers/eserver/zseries/os/linux/ifl.html>

■ HP vende notebook com Linux



Para clientes nos Estados Unidos, a empresa Hewlett-Packard está oferecendo o notebook modelo nx5000 com a opção do SuSE Linux 9.1 pré-instalado (a máquina também é vendida com o Windows XP). Com isso a HP torna-se o primeiro grande fabricante de notebooks a oferecer um equipamento atual com Linux. O nx5000 é um modelo de alto padrão (*business*), com um processador Pentium M, monitor LCD de 15 polegadas e WLAN na plataforma Centrino, disponível em diferentes configurações, e será vendido por um preço a partir de US\$1200. Interessante é que, até pouco tempo atrás, sistemas com chipset Centrino eram um desafio para instalações Linux, mas o SuSE Linux 9.1 vem com todos os drivers necessários para que o módulo WiFi Intel WLAN2100 (802.11b) e a placa de vídeo Intel Extreme Graphics 2 funcionem corretamente. ■

<http://h10010.www1.hp.com/wwpc/us/en/sm/WF05a/321957-64295-89315-321838-f33-395654.html>

■ CA abre código do banco de dados Ingres

A empresa Computer Associates (CA) está abrindo o código fonte do seu banco de dados, o Ingres. A CA espera com isso contribuições da comunidade para o desenvolvimento do sistema, bem como a sua utilização em outros produtos e projetos de código aberto.

Com esse propósito, foi criada uma licença própria, a “CA Trusted Open Source License” (CA-TOSL), que teve por base a “Common Public License” da IBM e que, como ela, deverá ser reconhecida pela Open Source Initiative (OSI). Bruce Perens, chefe da OSI, elogiou o retorno do Ingres às suas origens, que são livres: o projeto foi desenvolvido originalmente na Universidade da Califórnia, em Berkeley, durante a década de 70, tendo se tornado comercial apenas em meados dos anos 80, quando a CA o expandiu com recursos avançados e lhe deu confiabilidade comparável à dos bancos de dados comerciais. ■

<http://opensource.ca.com/projects/ingres/>

■ Governo e IBM assinam acordo para difundir o uso do Linux

No dia 31 de agosto, a IBM e o Governo Federal assinaram um contrato de cooperação para instalar um Centro de Difusão de Tecnologia e Conhecimento (CDTC), com a função essencial de promover e desenvolver soluções Linux e de software livre em geral no Brasil.

Sob os auspícios do Instituto Nacional de Tecnologia de Informação (ITI), com apoio do Ministério da Cultura e da Universidade de Brasília (UnB), o centro trabalhará para encorajar a popularização de soluções de padrão e código abertos através de treinamentos de técnicos, analistas de suporte e usuários dos sistemas da administração federal.

Situado na UnB, o centro irá apoiar desenvolvimento colaborativo e projetos de migração tecnológica de instituições públicas e microempresas, envolvendo o uso de soluções baseadas em padrões abertos. A administração do CDTC ficará por conta da IBM e do ITI, e sua primeira tarefa será o treinamento de 700 profissionais dos núcleos de tecnologia educacional (NTE) do Ministério da Educação. Há 325 destes núcleos espalhados pelo País. Eles prestam suporte educacional e

pedagógico para a rede de escolas públicas, são formados por cinco pessoas em média, e incluem, invariavelmente, profissionais da área de TI. Além de IBM e ITI, há uma empresa nacional, especializada em capacitação de pessoal em Linux, chamada 4Linux, envolvida no projeto. A empresa será responsável por ministrar os cursos em todas as capitais brasileiras, com objetivo de esclarecer todas as possibilidades e benefícios de modelos baseados em software livre e padrões abertos.

Para o presidente do ITI, Sérgio Amadeu, a parceria com a IBM é uma oportunidade a mais para o governo incentivar a adoção da tecnologia de código aberto em sua estrutura. “O centro de difusão permitirá a intensa disseminação do conhecimento pelo País, além de ampliar a inteligência tecnológica nacional”, afirma. Neste mês, o centro começa, também, a organizar seminários para funcionários especializados em TI de todos os Ministérios.

“Para países em desenvolvimento como o Brasil e governos focados em incentivar o desenvolvimento econômico e a diversificação da indústria de tecnologia da informação no país, os padrões abertos podem desempenhar um importante papel na estratégia governamental”, diz Rogério Oliveira, Presidente da IBM Brasil. “Uma estratégia, aliás, implementada por mais de 60 países do mundo e mais de 22 estados americanos, que já adotaram o open source.” ■

<http://www.iti.br/twiki/bin/view/Main/PressRelease2004Aug30B>

■ Microsoft formaliza temor ao Linux

Diante dos seus grandes lucros e imensas reservas financeiras, ninguém pensaria na Microsoft como uma empresa que enfrenta dificuldades financeiras. Para que isto permaneça assim, a própria Microsoft trata de descrever os riscos que podem ameaçar os seus negócios. Em seu balanço anual, enviado para a Comissão de Norte-Americana Valores Mobiliários (a SEC), a Microsoft explica qual é o risco de a empresa correr por causa do Linux – e reafirma a estratégia de driblar a concorrência com investimentos em “inovação, interfaces mais simples e maior compatibilidade”.

Ainda segundo a empresa, não é apenas a quase estagnação do crescimento de vendas de computadores pessoais que ameaça o sucesso dos negócios da fabricante de software: a concorrência do software não-comercial, Software Livre e Open Source, também pode influenciar negativamente o andamento dos negócios da corporação. “Nossa concorrência direta compreende empresas que seguem modelos de negócios alternativos ao modelo de exploração comercial de software.” Com um pouco de desdém a Microsoft descreve sua visão deste modelo alternativo: “As empresas atendem aos clientes com software de código aberto a custos baixos e faturam adicionalmente com serviços e produtos, sem ter que arcar com os custos de pesquisa e desenvolvimento do software.”

A empresa ainda não vê o Linux como uma grande ameaça para a sua hegemonia no desktop, devido à simplicidade de uso da interface do Windows e ao seu melhor suporte a hardware, conforme afirmam. De outro lado, ela leva a sério a concorrência na área de servidores e bancos de dados. Como a participação de distribuições Linux no mercado de servidores cresceu mais do que a de produtos Windows, a empresa teme que as vendas de produtos Microsoft possam cair, caso a aceitação de produtos e soluções de código aberto no mercado continue a crescer no futuro. “O crescimento das distribuições Linux no mercado reflete o apoio de instituições públicas (i.e., governo) à introdução de software de código aberto tanto no mercado de servidores quanto no de desktops.”

No geral, a Microsoft parte do pressuposto de que a popularidade crescente de um modelo de software não comercial representará um desafio também no futuro – principalmente devido às tentativas dos defensores de tais modelos em convencer governos do mundo todo a migrar para Software Livre. “Na medida que o software de código aberto ganhe mais aceitação no mercado, as vendas dos nossos produtos podem diminuir e poderemos nos ver forçados a reduzir nossos preços, o que pode conseqüentemente afetar nosso faturamento e reduzir nossas margens de lucro”. ■

<http://www.sec.gov/Archives/edgar/data/789019/000119312504150689/d10k.htm>

Entrevista com Eduardo Lopez

Adoção total do Linux

A gigante dos bancos de dados Oracle adota firmemente a plataforma Linux no desenvolvimento de soluções, nos processos internos e nas propostas aos clientes, como afirma o vice-presidente da empresa no Brasil, Eduardo Lopez

POR ALEXANDRE BARBOSA

A Oracle é um dos maiores provedores de ferramentas corporativas no mundo, fornecendo de bancos de dados a ferramentas de gestão empresarial (ERP). E entre os grandes fabricantes de soluções tecnológicas para empresas, a Oracle pode ser encaixada entre os grandes promotores do Linux, ao adotar a plataforma em seus sistemas internos e nos produtos recomendados aos clientes, como a família 10g, a mais recente geração de produtos Oracle, que abrange banco de dados, servidor de aplicações e gerenciador de sistemas corporativos (Oracle Enterprise Management). Lançada em novembro de 2003, essa família de produtos foi desenvolvida para tirar o máximo proveito das vantagens da plataforma Linux.

O investimento da Oracle na plataforma Linux vem desde 1998, quando a empresa criou uma equipe de desenvolvimento focado nesse sistema operacional. Naquele ano, a Oracle lançou a primeira versão do seu banco de dados compatível com Linux. Hoje, mais de mil desenvolvedores Oracle usam esse ambiente para desenvolvimento. Outra demonstração da força do Linux são os mais de 4 mil fornecedores de software independentes (ISVs) com os quais a empresa mantém parcerias, que optaram por desenvolver soluções com Linux. Na ponta dos serviços, a Oracle oferece suporte técnico integrado ao Red Hat Advanced Server e a todos os sistemas operacionais baseados na plataforma UnitedLinux.

Para saber um pouco mais sobre as estratégias da empresa, a equipe da Linux Magazine conversou com Eduardo Lopez, 44 anos e vice-presidente de Tec-

nologia da Oracle no Brasil, executivo que afirma que o Linux é uma plataforma recomendável tanto do ponto de vista da segurança como em desempenho, mesmo em servidores de missão crítica. Na entrevista exclusiva dada pelo executivo, Lopez falou sobre a vantagem do Linux poder romper a dependência de um único fornecedor de sistema operacional e acrescentou que a substituição dos modelos atuais de computação pelo desenvolvimento compartilhado pode ajudar a trazer inovações.

Linux Magazine » Há uma estratégia comercial específica da Oracle para Linux?

Eduardo Lopez » Nós não temos uma iniciativa segmentada para o Linux. O mais correto seria dizer que o Linux é um elemento-chave de nossa estratégia comercial. Isso significa que a Oracle, ao se aproximar de um cliente ou cliente em potencial, não visa apenas a venda de produtos para atender a um determinado problema. Nossa meta é mais ousada: queremos chegar ao cliente e mostrar a possibilidade de oferecer uma nova visão sobre o papel da tecnologia dentro da organização, sempre de forma alinhada com o negócio do cliente. E esse posicionamento implica em propor novas arquiteturas para a área de TI, envolvendo o emprego avançado de servidores de aplicações e ferramentas de colaboração, ou seja, propostas de soluções abrangentes que resultem em redução de custos e melhoria dos níveis de serviço. É aí que o Linux entra, porque ele é um elemento essencial para garantir alta disponibilidade. E isso não se traduz apenas em sistemas mais estáveis ou abrangentes mas também em novas abordagens, como a computação



em grid, que introduz uma nova forma de aproveitamento de equipamentos legados. Então podemos dizer que o Linux está dentro de uma visão de arquitetura tecnológica na Oracle, o que faz dele um elemento presente em todas as nossas apresentações.

LM » Mas a empresa mantém suas ofertas para outros sistemas operacionais?

EL » Sim. Mas isso não entra em conflito com a nossa proposta, que é centrada na arquitetura da tecnologia. O Linux faz parte de diversas soluções que podem ser propostas para um cliente corporativo, seja ao propor uma visão de alta disponibilidade ou uma solução mais eficiente de recuperação de dados. Mas falar só desses atributos ou de *disaster recovery* não é suficiente para motivar um executivo de tecnologia a pensar em mudar o ambiente tecnológico de Unix ou Microsoft para o Linux. O cliente precisa ter uma visão clara dos ganhos que ele pode obter com a evolução de sua infra-estrutura tecnológica e é por isso que insisto na importância da arquitetura dos sistemas, que deve ser pensada de forma a atender às necessidades imediatas, com bons índices de aproveitamento e preparada para o futuro.

LM » Qual é papel dos parceiros e integradores nesse processo?

EL » É fundamental contar com a participação dos parceiros e integradores. Primeiro porque, por maior que seja a estrutura da empresa, é difícil contar

com recursos para cobrir todo o mercado. Segundo, porque é através do trabalho conjunto com outros fornecedores que podemos propor aos clientes a melhor solução. Existem integradores que sabem lidar com todos os meandros do hardware, de sistemas operacionais, aplicações e bancos de dados, e muitas vezes é através dessas parcerias que conseguimos detectar oportunidades comuns e propor aos clientes novas formas de executar suas aplicações, obtendo vantagens no rendimento, escalabilidade ou redução de custos.

LM » Quando a empresa começou a desenvolver versões de seus aplicativos para o ambiente Linux?

EL » Nós começamos a olhar o Linux mais de perto a partir de 1998, seguindo o processo de evolução da plataforma. Ao perceber do que se tratava, a Oracle passou a apostar muito no Linux, desenvolvendo produtos que usam esse ambiente operacional como base para o processamento de informações. A empresa também fez grandes investimentos no sentido de portar suas aplicações para ambientes código aberto, o que foi completado com a versão 9 de nosso sistema, totalmente preparado para o Linux. Além disso, a empresa participa dos principais fóruns de desenvolvimento de soluções open-source.

LM » Que tipo de esforço a empresa faz em desenvolvimento?

EL » A Oracle conta com um grupo próprio - o Oracle Linux Kernel Group - com programadores especializados que trabalham em esquema 24x7, totalmente dedicados a desenvolver novos recursos ou promover melhorias em produtos envolvendo o Linux e até mesmo no próprio núcleo do sistema. Este time está dividido em duas equipes que ficam em núcleos nos Estados Unidos e na Índia. Além disso, a Oracle conta, hoje, com mais de 9 mil desenvolvedores certificados que podem criar novas ferramentas baseadas em nossos produtos e no Linux. Temos também uma ampla base de conhecimento técnico sobre Linux que fica à disposição do público interessado através do Oracle Linux Development Center, que é um centro de excelência que dá acesso a ferramentas, código fonte e versões de avaliação de nossos produtos. Gostaria de aproveitar e indicar dois endereços para os leitores

da Linux Magazine: o do Linux Technology Center (<http://otn.oracle.com/linux>) e o do grupo de desenvolvimento dedicado a fazer produtos da Oracle terem mais performance em ambientes Linux (<http://oss.oracle.com>).

LM » A empresa adotou uma estratégia de fornecer serviços de suporte para o sistema operacional em vendas envolvendo Linux. Essa oferta foi importante para fechar negócios no Brasil?

EL » A Oracle adotou uma política de, ao vender uma solução baseada em Linux, fornecer suporte técnico de primeiro nível sobre o sistema operacional para os clientes. Esse é um item importante, pois fornece um nível de tranquilidade maior para o cliente, mas não posso dizer que seja algo fundamental em termos de geração de negócio. É um serviço adicional que a empresa fornece para aumentar a segurança na escolha do cliente.

LM » Como a empresa encara o uso do Linux para aplicações de missão crítica?

EL » A Oracle é uma adota a política de utilizar internamente sua própria tecnologia. Muitos dos nossos sistemas corporativos rodam sobre Linux, a partir de um data center mundial que centraliza nossas operações. Vários de nossos sistemas são críticos para o negócio, como sistemas de administração interna de operações ou nosso sistema de e-mail.

Em Austin, no Texas, onde mantemos nosso datacenter, 100% dos cerca de 700 servidores de camada intermediária rodam Linux. Sobre essa plataforma, a Oracle mantém a maior parte dos sistemas da Oracle University, de e-mail, do website, assim como das camadas intermediárias dos aplicativos de ERP/CRM e o Oracle File System. Além disso, a maioria esmagadora dos 350 clientes do datacenter nos Estados Unidos também adota a plataforma.

LM » Que tipo de cliente da Oracle no Brasil opta pelas versões Linux? Há um perfil definido dessas empresas?

EL » Não há um perfil definido para estas empresas. A escolha do Linux entre os clientes da Oracle depende muito do papel da tecnologia da informação dentro da empresa. Mas há um número significativo de projetos envolvendo Linux em desenvolvimento no setor governamental, que é o segmento que está influenciando muito o desenvolvimento e a adoção de soluções open-source.

LM » Existem barreiras que dificultam a adoção do Linux? O número relativamente pequeno de profissionais certificados é um problema?

EL » A Oracle tem forte atuação em serviços de suporte ao cliente. Até agora não encontrei uma empresa que dissesse que não vai adotar Linux por falta de pessoal especializado. A escolha entre adotar ou não o Linux tem mais a ver com o desenho e a evolução da companhia e de seus conceitos internos.

LM » Qual é a próxima onda tecnológica envolvendo o Linux?

EL » Uma tendência que acho importante destacar para os próximos anos é a adoção das soluções de grid computing, para as quais o Linux é um ambiente muito interessante. Mas para adotá-las, as empresas precisam começar a repensar, desde já, a sua cultura interna em relação à tecnologia e a arquitetura de seus sistemas. O desenho dessa arquitetura, feito com perspectivas de adoção de soluções de grid, pode ser feito aos poucos. Já fizemos estudos que comprovam que a adoção de soluções de grid pode manter a mesma qualidade de serviços com reduções entre 20% e 25% nos custos com as arquiteturas.

LM » Qual é, na sua visão, o estado de maturidade do Linux? Por que as empresas ainda resistem a adotar o Linux?

EL » O Linux já provou seu valor. É uma plataforma madura e em contínuo desenvolvimento. Quanto a uma resistência na adoção, insisto que é uma questão de cultura interna e de arquiteturas. Em geral, o uso do Linux segue um comportamento padrão: a princípio a empresa usa o Linux em sistemas menos críticos como servidores web ou de e-mail; com o tempo, a confiança gerada leva a experiências com outros sistemas até chegar ao ponto em que a credibilidade permite a adoção do Linux em sistemas críticos. Recentemente um executivo de um grande cliente brasileiro visitou nosso datacenter em Austin. Lá, ele viu que estávamos fazendo o processamento de um sistema crítico de outro cliente, da Ásia, em regime de outsourcing, envolvendo cerca de 3 terabytes de dados. Foi quando ele percebeu que podia usar tranquilamente soluções da Oracle com Linux. Ou seja, é uma questão de tempo para que as empresas passem a enxergar o Linux de forma diferente. ■

Dicas de (In)Segurança

■ Subversion

O Subversion é um sistema de controle de versão similar ao popular CVS.

O código do subversion é vulnerável a um estouro de buffer na pilha que pode ser remotamente explorado. Este erro acontece antes de qualquer processo de autenticação. Um agressor pode usar esta vulnerabilidade para executar um código qualquer.

O projeto “Common Vulnerabilities and Exposures” deu a este problema o código CAN-2004-0413.

Código de Referência no Suse:

SuSE-SA:2004:018

Código de Referência no Gentoo:

GLSA 200406-07 / Subversion

■ Kernel

Recentemente foram descobertas múltiplas vulnerabilidades de segurança no kernel do Linux.

Uma falha nas versões 2.4 e 2.6 do kernel para as arquiteturas x86 e x86_64 permite que usuários locais causem uma negação de serviço (derrubem o sistema) ao ativar um receptor de sinal com uma certa seqüência de instruções fsave e fstor. O projeto “Common Vulnerabilities and Exposures” deu a este problema o código CAN-2004-0554.

Outra falha foi descoberta na chamada de sistema clone(). Ela permite que usuários locais causem uma negação de serviço (vazamento de memória) ao passar argumentos inválidos para clone() dentro de um loop infinito em um programa. O projeto “Common Vulnerabilities and Exposures” deu a este problema o código CAN-2004-0427.

Al Viro fez melhorias no kernel 2.6 que permitem que a ferramenta de checagem de código esparsa procure por certos tipos de bugs no kernel. Vulnerabilidades no acesso à memória do kernel nos drivers e1000, decnet, acpi_asus, alsa, aito/WLAN, pss e mpu401 foram corrigidas. Estas vulnerabilidades podem levar à leitura e escrita da memória do kernel e negação de serviço local, resultando na possibilidade de um agressor

com uma conta local ter acesso à conta de root do sistema. O projeto “Common Vulnerabilities and Exposures” deu a este problema o código CAN-2004-0495.

Um vazamento de informações que afeta apenas os sistemas ia64 também foi descoberto. O projeto “Common Vulnerabilities and Exposures” deu a este problema o código CAN-2004-0565.

Código de Referência no Mandrake:

MDKSA-2004:066

Código de Referência no Red Hat:

RHSA-2004:255-10

■ DHCP

O servidor DHCP (Dynamic Host Configuration Protocol) é usado para configurar dinamicamente clientes que se conectam a uma rede (“hotspots” wireless, redes empresariais, etc...)

Uma vulnerabilidade no modo como o servidor DHCPD do ISC (Internet Systems Consortium) lida com o registro das atividades do servidor via syslog pode permitir que um agressor derrube o daemon e cause um ataque de negação de serviço (Denial of Service - DOS) ao enviar pacotes especiais para a porta de escuta do DHCPD. Existe a possibilidade de execução de código com as permissões do usuário que está rodando o servidor, geralmente o root. A United States Computer Emergency Readiness Team (U.S. CERT) deu a este problema o código VU#317350.

Existe uma vulnerabilidade similar no modo como o DHCPD do ISC usa a função vsnprintf() em um sistema que não a suporta. Esta vulnerabilidade também pode ser usada para execução de código e/ou um ataque de negação de serviço. As declarações da função vsnprintf() afetadas estão logo depois do código mencionado no parágrafo anterior. A United States Computer Emergency Readiness Team (U.S. CERT) deu a este problema o código VU#654390.

Código de Referência no Suse:

SuSE-SA:2004:019

Código de Referência no Mandrake:

MDKSA-2004:061

■ SquirrelMail

O SquirrelMail é um sistema de webmail escrito em PHP. Foram encontradas múltiplas vulnerabilidades em uma das versões do software.

Uma falha que permite a injeção de comandos SQL foi encontrada no SquirrelMail 1.4.2 e versões anteriores. Se o SquirrelMail estiver configurado para armazenar o livro de endereços do usuário na base de dados, um agressor remoto pode usar esta falha para executar comandos SQL. O projeto “Common Vulnerabilities and Exposures” deu a esta falha o código CAN-2004-0521.

Também foram encontradas várias falhas no recurso “cross-site scripting” (XSS) no SquirrelMail 1.4.2 e versões anteriores, que permitem a um agressor remoto executar um script com os privilégios de outro usuário. O projeto “Common Vulnerabilities and Exposures” deu a este problema os códigos CAN-2004-0519 e CAN-2004-0520.

Código de Referência no Red Hat:

RHSA-2004:240-06

Código de Referência no Gentoo:

GLSA 200405-16 / SquirrelMail

■ Sup

O Sup (*Systems Utility Page*) é um aplicativo modular, escrito em PHP, que pode ser usado para monitorar e executar comandos em servidores locais ou remotos, além de sincronizar repositórios de arquivos. Pode ser encontrado no endereço: <http://freshmeat.net/projects/sup/>.

O usuário jaguar@felinemenace.org encontrou uma vulnerabilidade no sup que possibilita a um agressor remoto a execução de código com os mesmos privilégios de acesso do proprietário do processo *sup-filesrv*.

O problema está em uma string de formatação do syslog(3), encontrada nas funções logquit, logerr e loginfo. O projeto “Common Vulnerabilities and Exposures” deu a esta falha o código CAN-2004-0451.

Código de Referência no Debian:

DSA-521-1 sup — format string vulnerability

■ Libpng

O pacote libpng contém uma biblioteca com funções para criar e manipular imagens no formato PNG (Portable Network Graphics), uma alternativa livre ao popular formato GIF.

Durante uma auditoria nas atualizações da Red Hat, a equipe Fedora Legacy encontrou uma falha de segurança na biblioteca libpng que não havia sido corrigida no Red Hat Enterprise Linux 3. Um agressor pode criar um arquivo PNG que causa a queda de um aplicativo ligado à libpng, com a possibilidade de execução de código quando a imagem é aberta.

Foi descoberto um estouro de buffer causado pelo cálculo incorreto do valor de offset de um loop. Esta falha pode levar a um ataque de negação de serviço (DoS) ou mesmo comprometimento remoto do sistema.

Esta vulnerabilidade foi originalmente corrigida em janeiro de 2003, mas desde então descobriu-se que dois outros pontos do código também afetados não foram corrigidos. O projeto “Common Vulnerabilities and Exposures” deu a este problema o código CAN-2002-1363.

Código de Referência no Mandrake:

MDKSA-2004:063

Código de Referência no Red Hat:

RHSA-2004:249-07

Código de Referência no Gentoo:

GLSA 200407-06 / libpng

■ Apache

O servidor HTTP Apache é um servidor web completo, poderoso, eficiente e livremente disponível na Internet.

Um vazamento de memória que pode ser causado remotamente foi descoberto na versão 2.0.50 do servidor HTTP Apache. Isto possibilita que um agressor remoto cause um ataque de negação de serviço (DoS) ao fazer o servidor web consumir grandes quantidades de memória. O projeto “Common Vulnerabilities and Exposures” deu a esta falha o código CAN-2004-0493.

George Guninski descobriu uma outra falha que pode levar a um ataque de negação de serviço (DoS) no Apache 2.x. Esta falha também pode fazer com que o servidor web consuma mais memória que o normal. Em sistemas de 64 Bits com mais de 4 GB de memória virtual, ainda raros, isto também pode levar a um estouro da pilha.

Um estouro de buffer também foi encontrado por George Gudinski no módulo mod_proxy do Apache. Esta falha pode ser explorada por um usuário remoto para potencialmente executar código com os privilégios de acesso do proprietário do processo-filho do httpd (geralmente o usuário apache ou www-data). Contudo, esta falha só pode ser explorada se o módulo mod_proxy realmente estiver em uso.

Note que este bug existe em um módulo no pacote apache-common, compartilhado pelo apache, apache-ssl e apache-perl, de forma que uma única atualização é suficiente para corrigir estas três variantes do servidor httpd. Contudo, em sistemas usando o apache-ssl ou apache-perl, o daemon httpd não será reinicializado automaticamente. O projeto “Common Vulnerabilities and Exposures” deu a este problema o código CAN-2004-0492.

Código de Referência no Mandrake:

MDKSA-2004:064 e MDKSA-2004:065

Código de Referência no Red Hat:

RHSA-2004:342-10

Código de Referência no Debian:

DSA-525-1 apache — buffer overflow

Código de Referência no Gentoo:

GLSA 200407-03 / Apache

■ Super

O Super é um programa que permite a usuários específicos executar comandos com privilégios de root, de forma similar ao sudo. Pode ser encontrado no endereço <http://freshmeat.net/projects/super/>.

Max Vozeler descobriu uma vulnerabilidade em uma string de formatação no super. Esta vulnerabilidade pode ser explorada por um usuário local para executar código com privilégio de root.

O projeto “Common Vulnerabilities and Exposures” deu a este problema o código CAN-2004-0579.

Código de Referência no Debian:

DSA-522-1 super — format string vulnerability

■ www-sql

O www-sql é um programa que permite a criação de páginas HTML dinâmicas através da inclusão de tags especiais, possibilitando a interação com bancos de dados SQL, como o PostgreSQL.

Ulf Härnhammar descobriu um buffer overflow no módulo www-sql. Ao explorar esta vulnerabilidade, um usuário

local pode executar código ao criar e processar uma página com o www-sql.

O projeto “Common Vulnerabilities and Exposures” deu a este problema o código CAN-2004-0455.

Código de Referência no Debian:

DSA-523-1 www-sql — buffer overflow

■ rlpr

O rlpr é um pacote que permite a sistemas remotos imprimir arquivos em uma impressora local.

O usuário jaguar@felinemenace.org descobriu uma vulnerabilidade em uma string de formatação no rlpr. Ao investigar esta falha, um buffer overflow foi descoberto em um trecho de código relacionado. Ao explorar estas vulnerabilidades, um usuário local ou remoto pode causar execução de código com os privilégios do proprietário do processo rlprd (no caso de um usuário remoto) ou root (no caso de um usuário local).

O primeiro erro é uma vulnerabilidade em uma string de formatação do syslog(3) na função msg() no rlpr. O projeto “Common Vulnerabilities and Exposures” deu a este problema o código CAN-2004-0393.

O outro problema é um buffer overflow também na função msg() no rlpr. O projeto “Common Vulnerabilities and Exposures” deu a este problema o código CAN-2004-0454.

Código de Referência no Debian:

DSA-524-1 rlpr — several vulnerabilities

■ Pavuk

Pavuk é um aplicativo usado para espelhar websites ou arquivos. Ele pode transferir documentos de servidores HTTP, FTP, Gopher e opcionalmente HTTPS (HTTP sobre SSL). O software pode ser encontrado no endereço <http://www.idata.sk/~ondrej/pavuk/>

Ulf Härnhammar descobriu uma vulnerabilidade no Pavuk, no qual uma resposta HTTP 305 grande demais enviada por um servidor malicioso pode possibilitar a execução de código com os privilégios do proprietário do processo pavuk.

O projeto “Common Vulnerabilities and Exposures” deu a este problema o código CAN-2004-0456.

Código de Referência no Debian:

DSA-527-1 pavuk — buffer overflow

Código de Referência no Gentoo:

GLSA 200406-22 / Pavuk

Notícias do Kernel

■ Rastreamento de patches

Um novo sistema de atribuição de patches começa a surgir, graças, em parte, às alegações da SCO. Segundo Linus Torvalds, que propôs sua criação, o objetivo é facilitar a tarefa de rastrear um patch durante toda sua existência, desde seu autor original, passando pelos responsáveis por drivers e subsistemas intermediários e pela cadeia de desenvolvedores - o que acaba levando ao próprio Linus e outros mantenedores do kernel.

O mecanismo, ainda em discussão, não poderia ser mais simples: cada desenvolvedor que envia um patch simplesmente adiciona a ele um pequeno texto, no qual se identifica e declara a "liberdade" de sua contribuição. Desta forma, para rebater um ataque como os da SCO, basta verificar os "atestados" referentes à cada patch, em vez de ter de mergulhar em antigos arquivos de listas de discussão, como é feito atualmente.

Os atestados seriam algo mais formal, indicando que o desenvolvedor escreveu o código e o enviou sob a licença especificada, ou que modificou um código que já havia sido devidamente licenciado, ou ainda que recebeu a contribuição de alguém que autorizou sua inclusão no kernel e que o código não foi modificado de forma alguma.

Logo que Linus fez a proposta, começaram a aparecer patches usando o novo sistema de atribuição, apesar da possibilidade de modificações futuras. Uma coisa é certa: os desenvolvedores do kernel aprovaram a idéia básica e estão realmente dispostos a adotar o método proposto por Linus, ou algo que atinja os mesmos objetivos.

Só nos resta esperar para ver se isto ajudará mesmo na defesa contra as acusações de infração de copyright no kernel.

■ Alvos de compilação

Com a criação do novo sistema de compilação do kernel (kbuild) na série 2.5, um alvo de compilação (*build target*) para geração de pacotes RPM foi adicionado ao código-fonte. Logo em seguida,

Wichert Akkerman postou um patch para produzir um build target que gera pacotes .deb (do Debian).

Infelizmente, Wichert postou seu patch bem no meio do desenvolvimento da série 2.6.0-test, quando Linus Torvalds e Andrew Morton haviam "congelado" a adição de novos recursos até que pudessem lançar a versão 2.6.0 e inaugurar uma nova série estável. Na época, Wichert foi informado de que deveria reenviar seu patch mais tarde, quando a série 2.6 estivesse mais estável e liberada para a adição de novos recursos.

Isso aconteceu em outubro de 2003. Em junho deste ano, Wichert reenviou seu patch e desta vez teve uma recepção muito mais calorosa. Sam Ravnborg, que estava trabalhando em alvos para a geração de "pacotes" nos formatos .tar.gz e .tar.bz2, disse que incluiria os patches de Wichert entre os seus quando os oferecesse para inclusão no kernel.

Parece portanto que, em breve, alvos para a geração de pacotes .deb (e outros formatos) chegarão à árvore do kernel 2.6. Assim como o alvo para pacotes RPM, eles são, na verdade, mais uma questão de conveniência, já que a mesma tarefa pode ser realizada com scripts simples. Contudo, são bem fáceis de implementar e manter e fornecem um recurso que muitos usuários desejam.

■ NTFS ganha mais suporte

Anton Altaparmakov tem progredido com o driver para sistemas de arquivos NTFS. Até agora não existia suporte à escrita, e para evitar a corrupção dos dados era necessário executar um utilitário após montar uma partição NTFS.

A função do utilitário agora está incorporada ao próprio sistema de arquivos e já é possível sobrescrever um arquivo com novos dados. Não é tão bom quanto o suporte completo à escrita, mas é um belo começo. Embora todos concordem que o driver NTFS nunca será a melhor opção para um sistema de arquivos sob o Linux, ele é uma boa opção para compartilhar discos entre o Windows e o Linux.

■ SMBIOS

O driver SMBIOS foi removido do kernel 2.6, para tristeza de David Mosberger, que recentemente escreveu o código para permitir seu uso em máquinas baseadas na arquitetura ia64. O motivo para a remoção do driver é um velho conhecido: tudo o que ele faz pode ser feito no espaço do usuário.

Infelizmente, como o próprio David mostrou, a técnica depende do */dev/mem*, do qual muitos desenvolvedores querem se livrar ou, ao menos, desencorajar seu uso. Até mesmo Michael Brown, autor original do driver SMBIOS, foi a favor de sua remoção. O próprio Michael decidiu usar a solução baseada em */dev/mem* no espaço do usuário e disse que não pretende manter um código que nem ele mesmo usa mais.

Havendo o próprio autor do driver se oposto à sua inclusão no kernel 2.6, fica difícil convencer Linus Torvalds ou Andrew Morton a mantê-lo. Contudo, se pessoas como David realmente fizerem questão, é possível que um novo mantenedor seja encontrado, ou que pelo menos o debate continue até que seja encontrada uma solução que satisfaça aos dois lados: a turma que quer evitar adicionar ao kernel recursos que podem ser implementados no espaço do usuário e a turma que acha que soluções baseadas no espaço do usuário são ruins e devem ser substituídas.

■ PC9800

O suporte à sub-arquitetura PC9800 provavelmente irá desaparecer do kernel 2.6 num futuro próximo. Os mantenedores trabalharam duro, com a ajuda de Alan Cox, para incluir seus patches na árvore oficial do kernel, mas em seguida desapareceram, pararam de trabalhar no código e de responder aos e-mails.

Embora pessoas como Andrew Morton considerem o hardware PC9800 obsoleto, ele ainda é rápido e poderoso o bastante para uso em servidores e outras tarefas úteis. Infelizmente, a falta de um mantenedor sela seu destino.

Jeff Garzik e James Bottomley esperam que alguém se ofereça para manter o código, ou ao menos para organizar patches que possam surgir, mas se ninguém for encontrado o código será definitivamente removido.

■ O fim de uma era

Com o código quebrado e sem um mantenedor, o sistema de arquivos UMSDOS corre o risco de ser removido da árvore do kernel 2.6. Contudo, ainda há uma certa relutância entre os mantenedores do kernel, já que o UMSDOS foi parte importante da história do Linux.

Nos velhos tempos, a forma mais fácil de experimentar o Linux sem ter que formatar o disco era usar o UMSDOS para instalar o sistema em uma partição FAT já existente. Desta forma, era possível carregar o Linux ou o DOS a partir da mesma partição.

Considerado um belo *hack* em sua época (do início a meados da década de 90), o UMSDOS possibilitou o primeiro contato de muita gente com o Linux. Seria uma pena ter de aposentá-lo; mas sem ninguém para cuidar dele, parece que é exatamente o que vai acontecer.

A decisão não afeta quem usa o Linux hoje em dia. O código sequer funciona, mas seria formidável vê-lo corrigido e saudável, como uma espécie de monumento aos velhos tempos.

■ Mantenedor perdido

John A. Martin desistiu de manter o arquivo CREDITS e não se sabe se alguém irá substituí-lo. Isso não significa que o arquivo CREDITS deixe de existir, pelo contrário. Como disse Adrian Bunk, modificações nesse arquivo geralmente vêm dos próprios desenvolvedores, como parte dos patches submetidos. O arquivo se tornou parte tão popular da cultura do Linux que praticamente não há mais necessidade de um mantenedor.

Foi Adrian Bunk quem apresentou o patch para que o nome de John fosse removido do arquivo MAINTAINERS (onde constava como mantenedor do arquivo CREDITS). O próprio John concordou com isso, embora tenha se oferecido para voltar a manter o arquivo, caso haja necessidade.

■ Mantenedor encontrado

Greg Kroah-Hartman já vem trabalhando no SysFS há algum tempo e finalmente concordou em aceitar o título de mantenedor, incluindo seu nome na entrada correspondente no arquivo MAINTAINERS. Geralmente os desenvolvedores ficam entusiasmados por ter seus nomes neste arquivo, mas Greg já era mencionado várias outras vezes, principalmente por seu trabalho com PCI e USB, entre outras áreas.

Ele não chegou a resistir à idéia de ser nomeado mantenedor do SysFS; na verdade só não se apresentara, ainda, como tal. Provavelmente teria esperado ainda mais antes de ser oficialmente “empossado”, mas recebeu um empurrãozinho de Christian Gmeiner, que tentava, sem sucesso, descobrir quem era o mantenedor do SysFS. Frustrado, decidiu perguntar aos membros da lista de discussão linux-kernel. A partir daí, Greg não pôde mais se manter em silêncio.

Em sua vida relativamente curta, o SysFS (que surgiu no final de 2002) provou ser um grande sucesso e se tornou a interface entre o espaço do kernel e o espaço do usuário preferida por muitos aplicativos e drivers. Embora o Linux ainda suporte controles de I/O (*ioctl*s) e o sistema de arquivos */proc*, seu uso é desaconselhado em novos drivers; estes devem dar preferência à interface SysFS, que tem código mais limpo, manutenção mais fácil e não tem o peso da compatibilidade retroativa.

■ Dando nome aos bois

Uma coisa que nunca tinha sido feita até agora, segundo Linus Torvalds, é juntar os nomes de todas as pessoas que contribuíram com o kernel Linux. Além de um exercício intelectual, isso seria útil para rebater alegações como as da SCO.

Por mais ridículas que elas possam parecer aos observadores, ainda assim leva um bom tempo para preparar uma defesa contra uma alegação de infração de copyright no kernel Linux; o próprio Linus tem de lidar com várias delas (embora não sem ajuda).

Agora, Peter A. Van Tassell deu início ao projeto Grokline (veja o site oficial em <http://www.grokline.net>), para coletar em um só lugar todos os dados relativos à autoria do código presente no Linux. Geralmente esta é considerada uma

tarefa impossível, como quando alguém sugere mudar a licença do Linux da atual GPL para uma licença similar, como a Open Software License.

Quando surge uma sugestão desse tipo, logo vem alguém lembrar que, para isso, seria necessário rastrear cada uma das pessoas que colaboraram com o kernel durante toda a história do Linux e lhes pedir permissão para mudar a licença de sua contribuição.

Como já houveram milhares de colaboradores ao longo dos anos, a idéia de mudar a licença geralmente morre neste ponto. Contudo, a primeira tentativa séria de conseguir dados completos sobre a autoria dos patches para o kernel está em andamento.

■ Novo livro

Mel Gorman publicou um livro sobre o gerenciador de memória virtual do Linux! O título é “Understanding The Linux Virtual Memory Manager”, publicado pela Prentice Hall como parte da série “Open Source” de Bruce Perens. Mel começou a trabalhar nesta documentação antes mesmo que houvesse qualquer possibilidade de publicá-la em forma de livro. Foi inspirado pela decisão de Linus Torvalds de substituir o subsistema de memória virtual dos primeiros kernels da série 2.4 por uma implementação completamente nova.

A decisão causou muita discussão entre os desenvolvedores do kernel, já que a série 2.4 deveria ser estável, sem mudanças drásticas tais como uma substituição de todo o subsistema de memória virtual. Entre outras reclamações, a falta de comentários, ou qualquer outra forma de documentação sobre o código, tornou difícil para os desenvolvedores contribuir com aquela área do sistema durante um bom tempo.

Foi em meio a este conflito que Mel decidiu vasculhar o código, analisar seu comportamento e fornecer não apenas um comentário linha-a-linha, mas também uma explicação lógica de seu funcionamento em ensaios bem organizados. Foi essa empreitada que agora se transformou neste impressionante livro, que trata do assunto a fundo, num esforço para deixar os desenvolvedores curiosos mais perto de poder realmente participar do desenvolvimento da VM ou, pelo menos, de apreciá-lo em todos os detalhes. ■

Não há mais motivos para temer o *fsck*

Um lar para seus arquivos

Os modernos sistemas de arquivos para Linux dispõem de um recurso conhecido como “Journaling”, um registro cronológico das alterações efetuadas em cada arquivo. Verificações de integridade de sistemas com journaling ocorrem de forma extremamente rápida. Além disso, partições JFS, ReiserFS e XFS quase não têm limite de tamanho.

POR JAN KLEINERT

Com a explosão da quantidade de dados a ser armazenada e os discos rígidos e sistemas RAID tornando-se cada vez maiores, as desvantagens do velho sistema de arquivos ext2 ficam cada vez mais evidentes. A verificação de integridade dos dados nesse sistema de arquivos toma um tempo que parece não ter fim. E se houver a necessidade de reparos, lá se vão horas de trabalho. Não bastasse isso, há os limites de tamanho de arquivos e partições - e os arquivos se fragmentam com o tempo, o que prejudica bastante o desempenho do sistema.

Tudo isso não deveria ser de admirar: o ext2 foi desenvolvido ainda na pré-história do Linux, como um substituto para o sistema de arquivos do Minix, este vindo da era dos disquetes. Essa é a razão pela qual a estrutura do ext2 não é mais capaz de atender às necessidades dos centros de dados modernos.

Moderninhos

Entretanto, existem alternativas modernas, para as quais as limitações descritas acima são “fichinha”, como você verá na página 19. Ext3, JFS, ReiserFS e o XFS dispõem do journaling, recurso que torna a recuperação do sistema após eventuais travamentos ou quedas de energia uma questão de segundos.

Realizamos diversos testes para saber se o journaling tem algum efeito negativo no desempenho dos sistemas de arquivos. Confira os resultados a partir da página 23. Um dos sistemas de arquivos - para o qual, alias, quase não há ferramentas e opções de ajuste fino (comprove no artigo da página 26) - apresentou desempenho surpreendente.

A chave do problema

Outro artigo mostra como instalar e configurar um sistema de arquivos (via dispositivo loopback) criptografado, nas distribuições Debian, Gentoo e Red Hat. A chave criptográfica, essencial para acesso aos dados, fica arquivada em um simples “chaveirinho” USB.

Arrastando a âncora

Muitos sistemas Linux rodam de maneira lenta desnecessariamente. As

distribuições tendem a configurar a transferência de dados entre o disco rígido e controladora IDE de modo extremamente conservador, o que resulta em desempenho aquém do esperado. Veja no artigo da página 36 como usar, de forma segura, o utilitário *hdparm* para botar um fim a este problema.

E para encerrar as matérias de capa desta edição, reservamos um tutorial de LVM (“Logical Volume Manager”) sobre RAID por software. ■

CAPA

Journaling19

Presente nos sistemas de arquivos Ext3, JFS, ReiserFS e XFS, o journaling permite que sistemas reinicializados acidentalmente voltem a operar rapidamente.

Análise de desempenho23

Nossa extensa análise mostra as qualidades e fraquezas de cada um dos principais sistemas de arquivos para o Linux.

Ferramentas26

Cada sistema de arquivos tem as suas. Veja também como algumas opções do *mount* podem afetar o desempenho.

Criptografia de bolso31

Veja como criar um sistema de arquivos criptografado, com a chave armazenada em um “chaveiro” USB.

Turbine seu HD36

Obtenha ganhos incríveis de performance utilizando corretamente o *hdparm*.

LVM sobre RAID38

Veja em nosso tutorial como utilizar LVM sobre RAID via software.



Fundamentos dos sistemas de arquivos com journaling

Bloco da Alegria!

Um sistema de arquivos pode ser representado como uma camada de minúsculos blocos sobre a superfície de um disco. No início esta camada está organizada, porque todos os dados estão dispostos de modo limpo, uns depois dos outros, em blocos contíguos. Entretanto, a ordem se perde assim que os dados são postos em movimento (gravados e regravados), surgindo assim um efeito chamado fragmentação. Os novos sistemas de arquivos para Linux lidam melhor com esse processo do que o tradicional sistema de arquivos Ext 2. Em um teste comparativo detalhado entre o Ext 2 e o ReiserFS [1], o ReiserFS se comportou nitidamente melhor no tratamento da fragmentação de dados.

As longas pausas para o café, outrora necessárias quando o programa de verificação de integridade (*fsck*) examinava um sistema de arquivos Ext 2, foram abolidas. Os novos sistemas de arquivos para Linux registram as mudanças no sistema de arquivos em uma espécie de catálogo, o *journal* e por isso eles são chamados de sistemas de arquivos com *journaling*. Graças ao *journaling*, o programa *fsck* já terá reparado inconsistências no sistema de arquivo, enquanto o *fsck.ext2* ainda estiver verificando se existem efetivamente erros a eliminar. Motivo: o *fsck.journal*, utilizado pelo *fsck*, examina somente os dados anteriores à última entrada no *journal*. Com isso, o computador fica pronto para uso mais rapidamente e a disponibilidade do sistema aumenta.

Tamanho É documento.

A alta disponibilidade é importante porque o volume de dados cresce sem parar e, com ele, também a capacidade do hardware. Os fabricantes de discos rígidos já superaram a barreira dos 300 GB e há tempos os sistemas RAID já administram numerosos terabytes (TB). De qualquer modo, os projetistas de sistemas de arquivos pensam muito além do hardware disponível hoje e criam sistemas

capazes de administrar eficientemente mais de um milhão de terabytes (um exabyte, 1 EB), conforme ilustra a Tabela 1. O XFS da SGI lidera, com um tamanho máximo teórico para o sistema de arquivos de 1,125 EB no Linux, um pouco à frente do ReiserFS, cujo limite está em 1 EB. A IBM segue em terceiro, com 32 PB (petabyte: 1 PB corresponde a 1024 TB). O Ext 3 está restrito às limitações impostas pelo kernel 2.6, com um tamanho máximo de 16 TB. A versão 2.4 pode administrar, no máximo, 2 TB.

POR JÖRG REITTER



capazes de administrar eficientemente mais de um milhão de terabytes (um exabyte, 1 EB), conforme ilustra a Tabela 1. O XFS da SGI lidera, com um tamanho máximo teórico para o sistema de arquivos de 1,125 EB no Linux, um pouco à frente do ReiserFS, cujo limite está em 1 EB. A IBM segue em terceiro, com 32 PB (petabyte: 1 PB corresponde a 1024 TB). O Ext 3 está restrito às limitações impostas pelo kernel 2.6, com um tamanho máximo de 16 TB. A versão 2.4 pode administrar, no máximo, 2 TB.

Se o espaço em disco se tornar insuficiente, o usuário pode aumentar o tamanho dos sistemas de arquivos com o sistema ainda em funcionamento. Isto é especialmente interessante em conjunto com um volume lógico (*LVM - Logical Volume Manager*) ou com RAID (*Redundant Array of Inexpensive Disks*) via software. O artigo “Jogue seus dados”, na página 38 desta edição, descreve o uso das duas técnicas para aumentar, de forma quase ilimitada, o espaço em disco. O limite é o seu bolso.

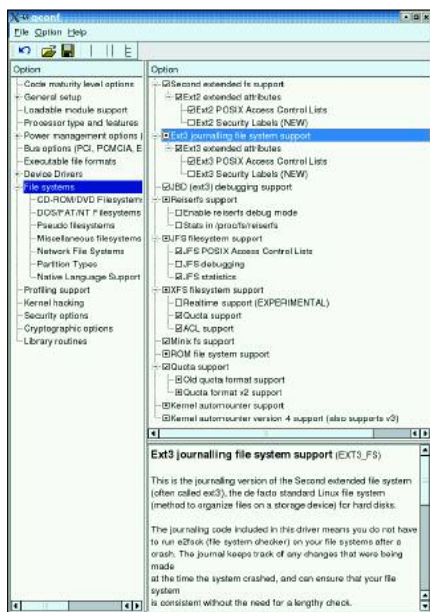


Figura 1: No kernel 2.6 há suporte para os sistemas de arquivos Ext 2/3, JFS, Reiserfs e XFS. As diferenças são no suporte a ACLs, quotas e funções de debug.

Sistemas de arquivos no kernel 2.6

O kernel 2.6 contém drivers para sistemas de arquivos Ext 3, JFS, ReiserFS e XFS, com os quais o usuário pode realizar testes confortavelmente, sem precisar alterar e recompilar as fontes originais. O kernel não só dispõe destes drivers, como também de funções modernas como suporte a quotas, listas de controle de acesso (“Access Control Lists”, as famosas ACLs) e suporte à “debugging” (depuração de erros), conforme mostra a Figura 1. O que permanece é o fato de que o suporte ao sistema de arquivos da partição *root* precisa estar compilado no kernel, caso um RAM disk inicial (*initrd*) não seja utilizado. Naturalmente, o usuário ainda pode compilar e carregar os sistemas de arquivos como módulos em tempo de execução.

No que concerne à administração de direitos do sistema de arquivos, os administradores de rede têm apelado claramente, nos últimos tempos, cada vez mais às ACLs. Elas expandem os padrões habituais de direitos do usuário (“user”, o proprietário de um arquivo), do grupo de usuários e outros, de modo que o administrador pode facilmente, por exemplo, liberar o acesso a arquivos em um servidor Samba para usuários Windows remotos. Para o usuário de Windows, os direitos de acesso aos arquivos

aparecem como se ele acessasse um servidor Windows NT/2000, ao invés de um servidor Samba. Veja mais sobre a configuração de ACLs em [2] e [3].

Dividir e conquistar

Nos servidores de arquivos há sempre problemas de espaço. As quotas são uma solução eficaz para evitar que o espaço em disco do servidor acabe. O administrador determina, para cada usuário, uma fatia fixa do sistema de arquivos. Os sistemas de arquivos Ext 2/3 usam o suporte a quotas padrão do kernel 2.6, ao contrário do XFS, que oferece suporte próprio. O desenvolvimento do JFS e do ReiserFS, entretanto, ainda não chegou tão longe: no sistema de arquivos da IBM, o suporte a quotas está no cronograma para versões futuras. O ReiserFS só terá suporte a quotas a partir de sua versão 4, mas quem precisar dele hoje pode aplicar um patch no kernel 2.4.

Com os programas utilitários temos um quadro similar: JFS e ReiserFS não trazem programas para “dump” e “restore”. O YaST2, do SuSE Linux, fornece, entretanto, uma interface de backup para o ReiserFS – contudo sem suporte a backups incrementais. Deste modo, os usuários devem obrigatoriamente se habituar ao uso dos aplicativos *tar* ou *cpio*. Os usuários do Ext 3 simplesmente utilizam os programas habituais do Ext 2 [4]. A equipe de desenvolvimento de sistemas de código aberto da SGI também criou um utilitário para backup próprio, disponível no pacote *xfsplogs* [5]. Naturalmente existem outros programas que também podem fazer backups incrementais e a área de TI das empresas devem sempre prever essa necessidade [6].

Rapidinho com journaling

Sistemas de arquivos com journaling contribuem muito para a alta disponibilidade no datacenter. Sistemas assim equipados recuperam-se e retornam à operação muito mais rapidamente após um crash do que sistemas de arquivos sem journal, como o Ext 2. Uma reinicialização não planejada do computador implica em risco de inconsistência de dados no sistema de arquivos, pois arquivos que estavam em uso naquele momento não foram corretamente fechados. Durante a reinicialização, o programa *fsck.ext2* precisa verificar todos os arquivos quanto à consistência, o que pode levar horas em grandes sistemas de arquivos. Isso já não acontece com sistemas de arquivos equipados com journaling. Durante a reinicialização, o programa *fsck.journal* examina somente os arquivos posteriores à última entrada no journal, de modo que o sistema torna-se operacional novamente em segundos (ao invés de horas).

Mas não é somente após um crash que sistemas de arquivos com journaling são superiores. No que tange à eficiência organizacional em nível mais baixo, eles superam o Ext 2 de longe: seu esquema organizacional se baseia em determinados algoritmos de busca e classificação, chamados “B-Trees” (ver Quadro 1: Árvores Balanceadas).

Em contraste com uma lista desorganizada, como a utilizada pelo Ext 2, em um sistema com journaling as referências aos blocos individuais, que representam um arquivo, são organizadas na forma de uma árvore. Esta estrutura é tão otimizada, que um processo precisa esperar no máximo a metade do tempo

Fragmentação: perda programada de performance

Um arquivo com tamanho de 10 kB está num sistema de arquivos com blocos com tamanho de 4 kB. O arquivo não cabe, portanto, num único bloco e o sistema de arquivos o distribui em três. A princípio, esses blocos estão em seqüência, um atrás do outro. Entretanto, em um sistema de arquivos muita coisa acontece: os usuários excluem e acrescentam novos arquivos e relatórios dos processos também aumentam seus registros. Com o alvoroço desta movimentação de dados, formam-se cada vez mais ilhas de blocos onde antes havia um sólido bloco de arquivos. O sistema de arquivos se fragmenta continuamente, o que faz com que o desempenho vá caindo pouco a pouco (ver também [1]).

Um programa de desfragmentação seria ideal para restabelecer a ordem. De qualquer maneira, a desfragmentação, com as dimensões dos discos atuais, só deve ser feita se houver muito tempo disponível para isso. Num ambiente de produção, é melhor formatar o disco novamente e restaurar o backup dos dados de volta para ele depois. Pouco importa se o sistema de arquivos tem 200 GB ou 20 TB de dados – cedo ou tarde a morte o atingirá.

que esperaria normalmente, até que o sistema de arquivos encontre os blocos pertencentes ao arquivo procurado.

Já que, vistas do ponto de vista da raiz, as ramificações da árvore apresentam sempre a mesma profundidade, o termo B-Trees transforma-se, no jargão da literatura técnica, em árvores balanceadas. Cada um dos três sistemas de arquivos com journaling que utilizam uma estrutura de arquivos baseada em árvores balanceadas usa uma variante diferente: uns são, assim, mais eficientes no tratamento de vários arquivos pequenos em um diretório, outros, no tratamento de arquivos muito grandes.

Um gargalo apresentado por todos os sistemas, com exceção do Ext 3, é o VFS (“Virtual Filesystem Switch”). Sua arquitetura foi desenvolvida em função de sistemas de arquivos baseado em inodes. Entretanto, o JFS, o ReiserFS e o XFS usam B-Trees ao invés de inodes. As B-Trees aceleram o procedimento de busca, mas criam também um problema: um único erro em sua estrutura já pode levar à perda de dados. Além disso, exportar tais sistemas de arquivos via NFS é uma operação problemática e, além disso, o JFS, o Reiserfs e o XFS não funcionam com as quotas padrão do kernel.

Ext 3 ou Ext 2 com journaling

Quem usa o Ext 3 [7] ou prefere o Ext 2 não assume riscos. Não somente a possibilidade de transformar um sistema Ext 2 num Ext 3 através do programa *tune2fs* é uma vantagem importante, mas também os programas de dump e restore são os mesmos. Em comparação com o Ext 2, o Ext 3 ganhou só a função de journaling, com três opções de configuração: *journal*, *writeback* e *ordered*.

Com a opção *ordered*, se um arquivo e seus metadados correspondentes são modificados, estes dados são, antes, gravados (“flush”) no disco. Este é o comportamento padrão. No modo *writeback* somente os metadados são gravados no journal – o arquivo é modificado somente durante a sincronização do sistema de arquivos (“sync call”). O modo mais seguro, e mais lento, é o modo *journal*, que protocola as modificações nos metadados e nos arquivos.

A grande desvantagem do Ext 3 é a implementação de diretórios, feita através de uma lista encadeada. Todos os

Quadro 1: Árvores Balanceadas

Com as árvores balanceadas, os sistemas de arquivos para Linux utilizam um conceito que é aplicado há mais de 30 anos às bases de dados. A distribuição regular (balanceada) dos dados pelo sistema de arquivos com journaling resulta num esquema de organização mais eficiente que a técnica em uso pelo sistema de arquivos Ext 2. O Ext 2 usa listas desordenadas, que são ineficientes para busca e exclusão de dados, uma vez que, no pior dos casos, o diretório inteiro precisa ser varrido em busca do arquivo.

Com as B-Trees, um esquema de ramificações entra na estrutura organizacional dos dados. Ramificações que provêm da mesma raiz têm sempre a mesma profundidade, de forma que o equilíbrio entre as operações realizadas na árvore pode ser garantido. Desta forma, a eficiência da busca em um grande volume de dados aumenta, uma vez que nunca é necessário varrer um diretório inteiro. Os dados são organizados numa estrutura de árvore, onde as folhas armazenam os dados, as ramificações (nós) contêm referências ao próximo nível da árvore ou, no final, às folhas.

Algoritmos otimizados levam à velocidade máxima

A B-Tree padrão armazena dados e chaves nos chamados nós internos e nos nós das folhas. Nó é um termo geral que designa um elemento da árvore, e pode significar a raiz, uma ramificação ou uma folha. O número máximo de nós que levam ao nó mais baixo (a folha) é denominado altura de uma árvore parcial (todos os elementos abaixo de um nó). A árvore é dita balanceada (ou em equilíbrio) se o número mínimo de nós existentes no caminho que leva da raiz de cada árvore parcial até uma folha diferir no máximo em 1 do número máximo de qualquer outro caminho. Uma explicação detalhada sobre as B-Trees pode ser encontrada no site da Linux-Fibel (www.linuxfibel.de/filesys.htm#supp2). O documento “Sorting and searching algorithms”, que se encontra em <http://epaperpress.com/sortsearch/download/sortsearch.pdf>, esclarece como funciona a implementação na linguagem de programação C.

outros sistemas de arquivos com journaling se baseiam em árvores balanceadas (ver Quadro 1: Árvores Balanceadas). Listas encadeadas apresentam um desempenho bem pior. A performance do Ext 3 piora nitidamente com o aumento no volume de arquivos, e quando se trata de encontrar arquivos. O Ext 3 também é fraco no quesito fragmentação externa. Uma análise deste comportamento por um longo período pode ser encontrada em [1].

Comparado com outros sistemas de arquivos com journaling, o Ext3 também fica pra trás em casos como arquivos muito grandes (> 1 TB), partições muito grandes ou muitos arquivos em um diretório. A razão para isso é que as estruturas internas apresentam muito poucos bits para lidar bem com tais casos.

JFS: subnutrido

O JFS [9] da IBM visa servidores de bancos de dados, servidores de arquivos e qualquer outra aplicação com grande necessidade de espaço em disco rígido. Além disso, o desenvolvimento da IBM valoriza principalmente a segurança dos dados e a escalabilidade do sistema. Apesar disso, o desenvolvimento do JFS para Linux está um pouco atrás do de

seus concorrentes. A lista de coisas a ser feitas é a mais longa entre eles e o sistema é o que traz menos opções. Mesmo assim, uma partição formatada em JFS já pode ter seu tamanho aumentado durante a operação regular do sistema, que também suporta as ACLs Posix.

O usuário pode analisar problemas no sistema de arquivos ativando *JFS debugging* na configuração do kernel. As mensagens de erro geradas serão, assim, escritas no *syslog*. A IBM também pensou nos administradores que gostam de estatísticas: com a opção adequada ativada na configuração do kernel, um conjunto de estatísticas a respeito da utilização do sistema pode ser encontrado em */proc/fs/jfs/*.

A organização de diretórios no JFS foi muito bem projetada. Pequenos diretórios, com no máximo oito arquivos, são gravados pelo sistema diretamente no inode. Todos os diretórios com mais arquivos são organizados em uma B-Tree. Como é um sistema de arquivos de 64 bits, O JFS é predestinado para a utilização com arquivos e partições “gigantes”. As ferramentas de administração para o usuário estão reunidas no pacote *jfsutils*. Para os kernel 2.6 o usuário vai precisar no mínimo da versão 1.0.14.

ReiserFS: preferência por “arquivinhos”

Há tempos se diz que a quarta versão do sistema de arquivos de Hans Reiser [8] logo fará a alegria dos servidores FTP deste mundo. Vamos reservar para uma futura edição da Linux Magazine uma análise do desempenho do recém-lançado Reiser4, como a nova versão do sistema de arquivos é chamada. No presente artigo examinamos com cuidado os recursos das últimas atualizações da versão 3, mais especificamente o ReiserFS 3.6, que se encontra no kernel Linux atual. Quem quiser se manter informado sobre as novidades da versão 4 deve se preparar para participar de uma lista de discussão bem ativa.

O ponto alto do ReiserFS é a sua excelente performance quando se encontram no sistema muitos arquivos pequenos. Pequeno aqui significa menores que 1 kB. Arquivos com menos que 1024 bytes são muito comuns em servidores de “news”, razão pela qual administradores de sistemas recomendam o uso do ReiserFS para este tipo de serviço. Por outro lado, o sistema não apresenta um desempenho tão bom quando tem que acrescentar e excluir muitos arquivos, como acontece, no diretório */tmp*.

O ReiserFS manipula arquivos com grande eficiência no que diz respeito a aproveitamento de espaço em disco: ele não os aloca em blocos de tamanho fixo de 4 kB, como é tradição, mas em blocos cujo tamanho é exatamente o necessário. Adicionalmente, ele comprime a parte final de vários arquivos que não preenchem um bloco inteiro, e as grava juntas em um bloco separado (técnica chamada “tail packing”).

No quesito escalabilidade, o ReiserFS vai muito bem, obrigado: teoricamente, o sistema de arquivos pode ter até, no máximo, 1 EB (1 Exabyte), bem como pode-se trabalhar nele com arquivos da mesma ordem de grandeza.

XFS: o futuro é agora

O XFS, da SGI, [10] está nitidamente mais desenvolvido que o JFS ou o ReiserFS. Ele dispõe de várias opções para ajuste de desempenho, a documentação é muito boa e, para cada ferramenta contida no pacote *xfsprogs*, existe uma página de manual (*man page*) correspondente. A implementação de código aberto feita pela SGI para o kernel 2.6.0 já contempla inclusive ACLs (*Access Control Lists*). Além disso, o XFS é o único sistema de arquivos a oferecer um suporte próprio para quotas. O XFS trata as informações sobre quotas como metadados e as integra ao journal. Assim, uma alta consistência para as quotas fica garantida. Por último, o formato das quotas é compatível com o utilizado no IRIX, sistema operacional Unix da própria SGI, o que permite a migração de dados sem a necessidade de conversão do sistema de arquivos.

O XFS, tal como o JFS, é um sistema de arquivos de 64 bits, estando assim bem equipado para abrigar arquivos de proporções monstruosas. Teoricamente, o sistema de arquivos pode ter, no IRIX, até 18 EB (18 Exabytes, ou seja, 19.327 352.832 Gigabytes) e conter, por exemplo, apenas dois arquivos de 9 Exabytes cada. Não é de se admirar que ele seja o único sistema de arquivos equipado com journaling capaz de oferecer redimensionamento em tempo de execução.

Conclusões

À exceção do Ext 3, os sistemas de arquivos com journaling são tecnicamente bem semelhantes. Somente no atual estado de desenvolvimento de cada um deles há diferenças. Como um sistema de arquivos Ext 2 pode ser facilmente convertido em um Ext 3, e este, além disso, é compatível com o primeiro, o Ext 3 deverá ser a primeira escolha para muitos. O que falta ao sistema em desempenho, ele compensa em robustez. Quem, por outro lado, estiver procurando por um sistema de arquivos com escalabilidade enorme e alta performance, não tem como escapar do XFS. Uma gama completa de aplicações e sua boa documentação são vantagens adicionais do sistema de arquivos desenvolvido pela SGI. A utilização do ReiserFS 3.6 está indicada para casos em que muitos arquivos pequenos devem ser manipulados pelo sistema. Vamos ver o que a versão 4 do sistema nos reserva. O JFS ainda não chegou ao mesmo estágio que o dos outros sistemas de arquivos, de modo que o seu uso ainda não é recomendado em ambientes de produção. ■

Recursos dos sistemas de arquivos

Recurso	Ext3	JFS	ReiserFS	XFS
Journaling:Dados/Metadados	s/s	n/s	n/s	n/s
Journal Replay Kernel/Aplicações	s/s	n/s	s/n	s/n
Journal Interno/Externo	s/s	s/s	s/n	s/s
Diretórios baseados em B-Trees	n	s	s	s
Mapa dinâmico de alocação de inodes	n	s	s	s
Atributos estendidos/ACLs Posix	s/s	n/s	n/n	s/s
Quotas	s	Patch	Patch	s
Aumento/redução offline	s/s	n/n	s/s	s/n
Aumento/redução online	n/n	s/n	s/n	n/
Tamanho de bloco implementado (em kB)	1,2,4	4	4	4
Tamanho máximo do sistema de arquivos	16 TB	32 PB	1 EB	1,125 EB
Tamanho máximo dos arquivos	2 TB	4 PB	1 EB	562,5 PB

INFORMAÇÕES

- [1] Fragmentação: <http://www.informatik.uni-frankfurt.de/~loizides/reiserfs/thesis.html>
- [2] Curso sobre ACLs: <http://acl.bestbits.at/>
- [3] ACLs no SuSE: http://portal.suse.de/sdb/de/2002/10/81_ad.html
- [4] Dump/Restore para o Ext 2/3: <http://sourceforge.net/projects/dump/>
- [5] Xfsprogs: <ftp://oss.sgi.com/projects/xfs/>
- [6] Software de backup: <http://www.linuxmafia.com/faq/Admin/tape-backup.html>
- [7] Third Extended Filesystem (Ext3): <http://www.zip.com.au/~akpm/linux/ext3/>
- [8] ReiserFS: <http://www.namesys.com/>
- [9] JFS: <http://oss.software.ibm.com/jfs/>
- [10] XFS: <http://oss.sgi.com/projects/xfs/>
- [11] White Paper da Red Hat sobre o Ext 3: <http://www.redhat.com/support/wpapers/redhat/ext3/index.html>
- [12] Journaling Filesystems: http://en.wikipedia.org/wiki/Journaling_filesystem
- [13] Introdução ao Reiser4: <http://www.kuroshin.org/story/2003/8/9/172159/7912>

Ext 2/3, JFS, Reiser e XFS, quem ganha essa?

Provão!

Testes de desempenho de sistemas de arquivos são provas de fogo: servem para medir a velocidade de leitura e escrita de informações. A Linux Magazine colocou a performance do Ext 2/3, JFS, Reiserfs e XFS à prova.

POR MIRKO DÖLLE E JÖRG REITTER

Testes de desempenho são importantes indicadores de performance. Quando aplicados a sistemas de arquivos, eles indicam como o sistema reage a certas operações como leitura e escrita e o quanto é exigido do processador. Quem, por exemplo, armazena muitos arquivos pequenos em um diretório, necessita de um sistema de arquivos que vasculhe rapidamente o diretório e que se fragmente muito lentamente. Por outro lado, administradores de bancos de dados estão mais interessados em suporte a grandes arquivos e esperam uma excelente performance de leitura e escrita.

A Linux Magazine examinou o comportamento dos sistemas de arquivos Ext 2, Ext 3, JFS, ReiserFS e XFS com o conjunto de testes de desempenho Fsbench [1], que se compõe essencialmente de um script escrito em Python que chama os verdadeiros programas de comparação de desempenho: o Bonnie++ [2] e o Iozone [3], fornecendo-lhes os parâmetros apropriados (veja o Quadro "Ambiente de Testes").

Briga pela "pole"

Após a realização dos testes de desempenho, não nos causou espanto ver que o JFS foi, de longe, o vencedor. Entretanto, a versão para Linux do sistema de arquivos de código aberto da IBM traz claramente menos opções de ajustes que os outros sistemas (ver artigo à página 26). Contudo, ele se saiu bem em todos os testes, estando quase sempre entre os três primeiros. O segundo lugar foi um páreo duro, visto que os três concorrentes: o ReiserFS, o XFS e o Ext 2 travaram uma bela batalha pela colocação.

Finalmente, o consumo de CPU guiou a decisão, que deu o segundo lugar ao econômico XFS. O ReiserFS mostrou seu poderio com muitos arquivos pequenos em um mesmo diretório. Os bons valores de velocidade de escrita e leitura foram obtidos ao custo de uma carga de CPU razoavelmente grande. O uso do ReiserFS deveria, por isso, ter como pré-re-

quisito um hardware mais poderoso. Em vez disso, os administradores de sistemas devem optar pelo JFS ou XFS, que se saem bem melhor com hardware menos poderoso.

O Ext 2 continua a convencer, com uma performance apenas ligeiramente pior que a dos três sistemas citados até agora. Ele realiza todas as tarefas com desempenho entre médio e bom e não sobrecarrega muito o processador. O Ext 3, entretanto, apresentou valores muito ruins – ao menos em nossos testes. Motivo: configuramos o Ext 3 para operar propositalmente não em seu modo mais rápido (*ordered*), mas no que oferece mais segurança (*journal*). Assim, o Ext 3 atualiza primeiro os metadados e depois os escreve imediatamente no sistema de arquivos.

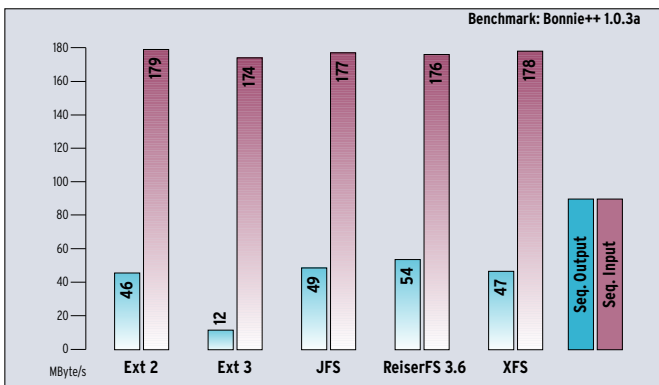
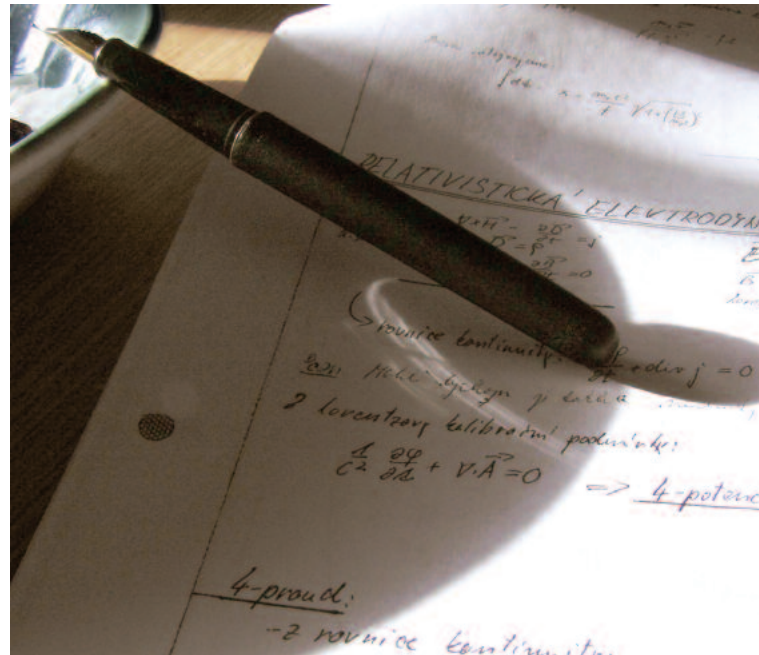


Figura 1: Taxas de transferência com arquivos de 1 GByte.

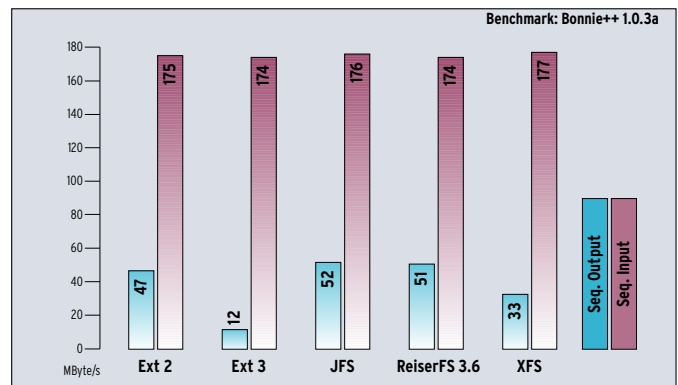


Figura 2: Taxas de transferência com arquivos de 4 GBytes.

Nos mínimos detalhes...

O Bonnie++ realiza diversos testes e os agrupa em saída seqüencial, entrada seqüencial, buscas aleatórias, criação seqüencial e criação aleatória de arquivos. A saída seqüencial mede a taxa de leitura por caractere e por bloco. Da mesma forma, o Bonnie mede o comportamento com a entrada seqüencial. Os gráficos mostram a performance de transferência de entrada e saída seqüenciais por bloco. Percebe-se imediatamente que o desempenho de leitura de todos os sistemas de arquivos são relativamente iguais, não importando se eles lidam com um arquivo de 1 GByte (Figura 1) ou de 4 GBytes (Figura 2). Totalmente diferente é a performance de escrita (Figura 3), na qual arquivos de 1, 2 e 4 GByte foram utilizadas nos testes.

O desempenho do Ext 2 ainda é bem semelhante ao do JFS, XFS e ReiserFS. O mesmo não se pode dizer do sistema de arquivos Ext 3. É uma surpresa que o modo journal emperre tanto o sistema de arquivos. Salta aos olhos a queda de performance com arquivos de 2 GB – problema que o ReiserFS também demonstra.

O sistema de arquivos JFS, da IBM, apresenta desempenho constante e, mesmo com tamanhos crescentes de arquivos, consegue um resultado melhor.

Com o XFS acontece precisamente o inverso. O sistema de arquivos da SGI é muito eficiente com arquivos de 1GByte, contudo perde consideravelmente em performance com arquivos de 2 e 4 GByte. Até mesmo o Ext 2 é nitidamente mais rápido com arquivos de 4GByte.

Os resultados alcançados pelo Bonnie com as buscas aleatórias não são mostrados nos gráficos. Nesta categoria do teste, o programa acessa blocos escolhidos aleatoriamente. Aqui o Ext 2 e o Ext 3 ficam claramente na frente e mostram, com todos os tamanhos de arquivos, performances de um a dez por cento melhores que os outros sistemas.

No teste de criação seqüencial de arquivos o Ext 2 é realmente bom: cria em média aproximadamente quatro vezes mais arquivos por segundo que os outros. No teste de leitura o Ext3 e o XFS brigam pela “pole position”. Somente no teste de exclusão de arquivos o ReiserFS demonstra que está bem acima da média e mostra em casos isolados um desempenho mais de 100 por cento melhor que o dos outros sistemas.

Ascensão e queda

O teste de criação aleatória mostra o mesmo quadro: o Ext 2 e o Ext 3 criam arquivos a uma taxa extremamente alta.

Do outro lado, na leitura e na exclusão de arquivos, o ReiserFS tem desempenho nitidamente melhor. O XFS fica bem atrás da concorrência nos testes de criação. Por outro lado, ele não chega a usar nem 0,1% do tempo de CPU, sobretudo durante o teste de exclusão de arquivos (o que ocorre também com o JFS) e com isso alcança resultados de 1% a 400% melhores que o Ext2, que consome entre 2 e 7 por cento do tempo do processador. E por falar em utilização de processador, neste quesito o ReiserFS faz feio: apesar de excluir arquivos muito rapidamente, ele utiliza no mínimo 7 por cento da CPU, chegando a usar até 99 por cento, no teste de criação aleatória: exclusão de 10.000 arquivos de 2 GByte, o que não é exatamente o desempenho ideal para um servidor em ambiente de produção.

Iozone: Desempenho de leitura e escrita

O benchmark Iozone cria arquivos com 1, 2 e 4 GByte e utiliza blocos de valores entre 64 bytes e 16 kBytes para os testes. Como na análise anterior, tanto a medida da performance de escrita quanto da de leitura eram de nosso interesse. A Figura 4 mostra a performance de escrita com arquivos de 2 GByte. Foram empregados

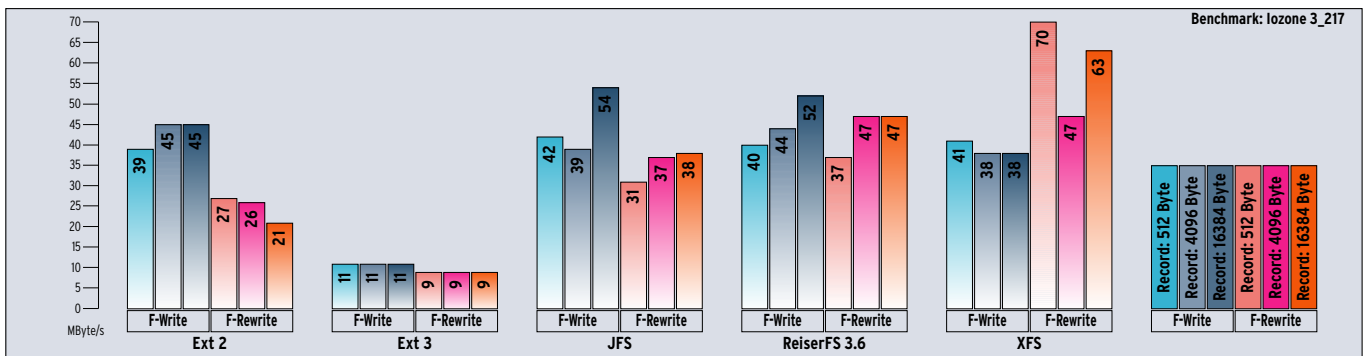


Figura 4: Desempenho de escrita com arquivos de 2 GBytes.

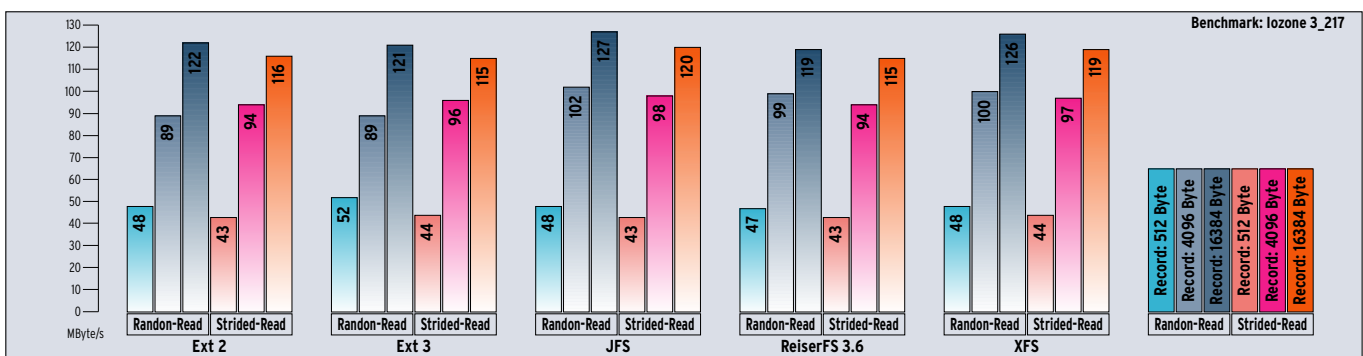


Figura 5: Desempenho de leitura com arquivos de 2 GBytes.

os testes F-Write e F-Rewrite nas medições. Estes utilizam a função `fwrite()` – da biblioteca `stdio` –, que escreve no sistema de arquivos nos modos “buffered” e “blocked”.

Com o teste F-Write, o Iozone cria um novo arquivo, e isso gera um trabalho extra de gravação dos metadados, o que prejudica o desempenho. Já o teste F-Rewrite escreve num arquivo já existente. O gráfico da Figura 4 ilustra muito bem este comportamento, ao menos para o XFS e o ReiserFS. O XFS bate de longe a concorrência em ambos os testes. O desempenho do JFS é relativamente equilibrado. Entretanto, ele obviamente trabalha melhor quanto maior for o arquivo. Contudo, a relação entre os desempenhos sob o F-Write e o F-Rewrite é desigual, o que ocorre também com o Ext 2, cuja performance sob o F-Rewrite despenca. Aqui, o Ext 3 cai novamente na categoria dos “lanternas”.

Os gráficos na Figura 5 mostram um desempenho homogêneo. Nela colocamos lado a lado os testes de leitura aleatória e leitura seqüencial intervalada (“strided”). O ouro olímpico nas duas modalidades vai para o JFS. As distâncias são, contudo, relativamente pequenas. Vê-se bem no gráfico as taxas de transferência crescentes quanto maior for o registro a ser lido. A princípio os resultados do Iozone no desempenho de leitura coincidem com os do Bonnie. Os dois indicam uma performance de leitura constantemente boa, indiferentemente se seqüencial ou aleatória e aplicada a pequenos ou grandes registros.

Outros comparativos de desempenho.

A Linux Magazine não é a única a realizar testes de desempenho em sistemas de arquivos. Usuários e os pró-

Ambiente de testes

O hardware que serviu de base para nossos testes foi o sistema de armazenamento conectado à rede Zero-One RM-250, da empresa alemã CTT (www.ctt.de), para usuários finais www.mailag.de), um sistema Pentium 4, rodando à 2,4 GHz, com 512 MByte de RAM, uma controladora RAID SATA modelo 8506-8 da empresa 3Ware e oito discos rígidos Maxtor, modelo 7Y250Mo (SATA). Cinco discos rígidos foram agrupados em uma configuração RAID5 com *hot-spare*, sendo que o sexto foi usado para a instalação do SuSE Linux Enterprise System 8 na sua configuração padrão (kernel 2.4.19-120).

O benchmark para sistemas de arquivos que foi utilizado vem de [1] e precisou apenas de ajustes mínimos para rodar no hardware de testes: primeiro os caminhos para os programas do benchmark (Bonnie++ e Iozone) não estavam corretos, depois o resultado do comando `mkfs.reiserfs -V`, que serve para indicar a versão do sistema de arquivos, retornou misteriosamente 1 ao invés de 0, o que foi corrigido utilizando-se `/bin/true`.

O programa Fsbench chama o Bonnie++ e o Iozone com os parâmetros correspondentes. Em nossos testes foi usada a versão 1.03a do Bonnie++ e a versão 3_217 do Iozone, ambos compilados com o a versão 3.2 do compilador GCC, presente no SuSE Linux Enterprise Server 8.

Foram utilizados 16 GBytes de espaço no sistema RAID, pois a controladora 3Ware usa fatias de dados de 64 KBytes, mas estes dados são distribuídos igualmente por todo o sistema RAID.



A máquina utilizada para os testes comparativos foi um Zero-One RM-250 da empresa CTT, com oito discos rígidos SATA e uma controladora RAID 3Ware.

prios desenvolvedores dos benchmarks publicam resultados na Web.

O site do Fsbench já aplicou a combinação Bonnie-Iozone ao kernel 2.6. Contudo, os resultados não diferem dos da Linux Magazine, que foram realizados com o Kernel 2.4.19, rodando no SuSE Linux Enterprise Server 8 (veja quadro “Ambiente de Testes”). O JFS e o XFS oferecem os melhores resultados de ponta a ponta. Para eles não faz diferença se os arquivos são grandes ou pequenos, nem se há muitos ou poucos deles em um diretório. Para a versão beta do ReiserFS 4 o resultado do Fsbench determina um aumento de desempenho de 65% sobre a versão 3.6, usada em nossos testes. Com isso o ReiserFS se

projeta no quesito eficiência, mas ainda assim, não importa a versão, é um consumidor voraz de tempo de processador.

O Ext 2 novamente se sai bem, tanto no que diz respeito à performance nos testes de escrita e leitura quanto na demanda de CPU, esta última pelo me-

nos duas vezes melhor que a do ReiserFS. Os resultados coincidem igualmente no que concerne ao Ext 3. No lento, porém seguro modo “journal”, ele fica bem atrás dos outros sistemas de arquivos. Assim que ele é configurado nos modos “ordered” ou “writeback”, quase não há mais diferenças entre os seus resultados e os do Ext 2.

O leitor encontrará outros testes comparativos nos sites da Guru-Labs [4], que comparam o Ext 3 com o ReiserFS 3.6, utilizando para isso o benchmark de código aberto *Postmark* da Netapps [5]. Este benchmark visa especificamente aplicativos como servidores de rede e email. Lá, o Ext 3 (em modo “ordered”) é o vencedor indiscutível. ■

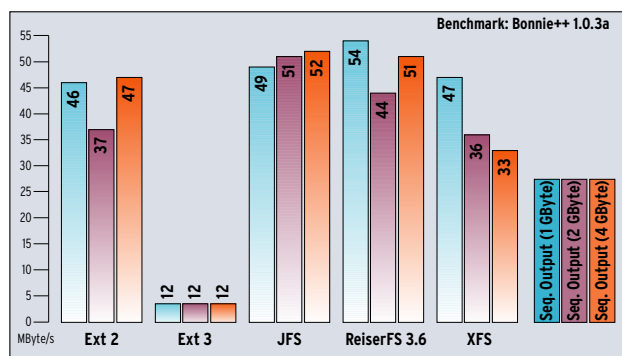


Figura 3: Taxas de transferência para arquivos de diversos tamanhos.

INFORMAÇÕES

- [1] Fsbench: <http://fsbench.netnation.com/>
- [2] Bonnie++: <http://www.coker.com.au/bonnie++/>
- [3] Iozone: <http://www.iozone.org/>
- [4] Guru Labs: <http://www.gurulabs.com/ext3-reiserfs-2.html>
- [5] NetApps: http://www.netapp.com/tech_library/postmark.html

Workshop: Ferramentas para manipulação de sistemas de arquivos

Oficina de formatação

A manutenção de sistemas de arquivos não é mais uma ação necessariamente demorada. Tanto o JFS, quanto o ReiserFS e o XFS lidam bem com grandes partições. O JFS e o XFS não precisam de mais que um piscar de olhos para formatar um disco rígido de 1 TByte, operação que o ReiserFS leva meio minuto para fazer. Apenas o Ext 3 ainda deixa o usuário esperando por cerca de uma hora, o que não deveria ser de admirar, afinal, no fundo ele continua sendo um sistema de arquivos Ext 2, equipado com recursos de journaling.

Além dos programas de formatação, existem ainda muitos outros utilitários para cada sistema de arquivos. Alguns deles podem ajudar a melhorar a performance. Outros são usados somente na ocorrência de erros. A seguir apresentamos os comandos mais importantes. Elas são utilizadas ou a partir da linha de comando com o comando *mount* ou de alterações efetuadas pelo usuário no arquivo */etc/fstab*.

Ext 2/3: rico em opções

É uma boa idéia equipar um sistema de arquivos Ext 2 com um journal, e atualizá-lo assim para Ext 3. Todo o processo é simples e não causa problemas; pelo menos os usuários nunca relataram nada nos fóruns específicos. Ainda assim, nunca é uma má idéia fazer um backup de todos os seus dados importantes antes da operação.

Para usar o Ext 3, o suporte a Ext 3 precisa estar ativado no kernel. Se você equipa a partição root com Ext 3, o driver do Ext3 deve ser, via de regra, compilado diretamente no kernel. Se se trabalha com um disco de RAM (initrd), entretanto, pode-se compilar o driver como módulo. A partir do momento em que o suporte a Ext 3 está disponível, a atualização pode começar. Para esta finalidade basta utilizar o programa *tune2fs*, contido no pacote *e2fsprogs* [1]:

```
tune2fs -j /dev/hda2
```

Os programas de formatação atuais fazem muito mais do que apenas criar sistemas de arquivos. Como caixas de ferramentas bem organizadas, eles oferecem um utensílio ideal para cada tarefa. Quem quiser aumentar o desempenho do seu sistema pode arrancar até o último bit de desempenho do Ext 3, JFS, ReiserFS ou XFS se utilizar corretamente as opções do comando *mount*.

POR JÖRG REITTER



A opção *-j* cria o journal e o coloca no arquivo *.journal* no diretório raiz da partição. Para concluir, modifique o tipo do sistema de arquivos de Ext 2 para Ext 3 no arquivo */etc/fstab*. Esta é uma das raras ocasiões em que excepcionalmente é mais simples reinicializar o sistema para que as modificações surtam efeito. Se depois você quiser converter o sistema de arquivos novamente em Ext 2, basta desmontá-lo e montá-lo novamente com o tipo *ext2*. Se quiser tornar isso permanente, modifique o tipo do sistema de arquivos novamente, alterando o arquivo */etc/fstab*.

Quanto maior o sistema de arquivos, mais tempo leva a verificação de integri-

dade, que acontece a intervalos pré-determinados. Caso os intervalos sejam muito espaçados ou curtos demais (por exemplo, como no caso de testes intensivos), este comportamento pode ser alterado de acordo com as necessidades do usuário. Com a opção *-i* do *e2fsck* ajusta-se os dias, semanas ou meses nos quais o programa deve executar a verificação de integridade do sistema de arquivos, que pode também depender do número de vezes que ele é montado – o parâmetro *-c* é usado para configurar o sistema desta forma. O argumento *0*, usado em ambas as opções, desativa completamente a verificação do sistema de arquivos.

Pode formatar!

Para começar, o ideal é realizar alguns testes em uma partição pequena. Sem opções adicionais, o comando de formatação `mkfs.ext3` determina os parâmetros necessários automaticamente. Assim, é relativamente simples formatar uma partição. Digite:

```
mkfs.ext3 /dev/sdb2
```

O tamanho de bloco apropriado é calculado pelo programa a partir do tamanho da partição. Se este não for o comportamento desejado, o próprio usuário pode fornecer o tamanho do bloco a ser utilizado (1,2 ou 4 kByte):

```
mkfs.ext3 -b 2048 /dev/sdb2
```

O `mkfs.ext3` oferece também opções com finalidades específicas. Com a opção `-T` o usuário ajusta o desempenho do sistema de arquivos de acordo com o tamanho dos arquivos a ser manipulados. Baseado nisso, o programa determina as configurações ideais. Os tamanhos disponíveis são 4 KBytes (*news*), 1 MByte (*largefile*) e 4 MBytes (*largefile4*). O comando a seguir parte do princípio de que a maioria dos arquivos do sistema de arquivos terão 1 MByte:

```
mkfs.ext3 -T largefile
```

Blocos malvados!

Nunca é demais verificar a integridade do disco rígido antes de instalar o sistema de arquivos. A opção `-c` instrui o programa `mkfs.ext3` a examinar o disco à procura de blocos ruins – os famosos “bad blocks”. Esta opção deve ser aplicada principalmente em hardware mais antigo. Para saber exatamente como o meio físico do disco está, pode-se especificar a opção `-c` duas vezes seguidas, o que instrui o `mkfs.ext3` a executar um teste (**destrutivo!**) de leitura e escrita (“read-write”), ao invés de somente um teste de leitura (“read”). Entretanto, este tipo de teste, além de arriscado, demora duas vezes mais para terminar.

O `mkfs.ext3` também conta com opções de journal. O usuário seleciona se o journal deve ser armazenado internamente no superbloco (`size = 1024`) ou externamente, em outro dispositivo (como em `device = /dev/hdc2`). No caso

Ext 3: opções do comando `mount`

Você encontra informações detalhadas sobre o sistema de arquivos Ext 3 em

```
/usr/src/linux/Documentation/filesystems/ext3.txt.
```

Nota: O Ext 2 ignora opções de quota como `grpquota`, `noquota`, `quota` e `usrquota` sem mostrar mensagens de erro.

data=journal: Os metadados vão primeiro para o journal, e somente então são escritos no sistema de arquivos (metadados e journaling de dados).

data=ordered: Primeiro os dados são escritos no sistema de arquivos; somente então os metadados são adaptados.

data=writeback: Os dados “reais” podem ser escritos no sistema de arquivos antes ou depois da atualização do journal de metadados.

journal=update: Atualiza o journal para o formato corrente.

journal=inum: Com `inum` o super-usuário fornece o número do inode que representa o arquivo do journal (ignorado, caso já exista um journal).

noload: O Journal não é carregado.

bsddf: O comando `df` reage como no BSD.

minixdf: O comando `df` reage como no Minix
check=none, nocheck: Não faz nenhuma verificação suplementar de bitmaps no decorrer da montagem.

debug: O kernel envia informações suplementares de depuração ao Syslog.

errors=continue: A montagem é levada adiante, mesmo que ocorra um erro no sistema de arquivos.

errors=remount-ro: No caso de erro, o sistema de arquivos é montado novamente apenas em modo de leitura.

errors=panic: A máquina pára no caso de erro no sistema de arquivos.

grpuid, bsdgroups: Dá a um objeto a mesma ID de grupo do objeto pai.

nogrpuid, sysvgroups: Objetos novos recebem a ID de grupo do criador.

resuid=UID: Aqui ficam as IDs de usuários que têm acesso a blocos reservados

resgid=GID: Aqui ficam as IDs de grupos que podem utilizar blocos reservados.

sb=offset: Utiliza um superbloco alternativo, encontrado na posição indicada em `offset`.

da variante externa, o journal já deve existir antes que o usuário crie o sistema de arquivos via `mkfs.ext3`. Neste caso, dois passos são necessários: em primeiro lugar devemos formatar a partição onde o journal ficará. Ela deve ter o mesmo tamanho de bloco (1, 2 ou 4 KBytes) do sistema de arquivos que vai utilizar o journal nela armazenado. O comando a seguir gera o journal para um sistema de arquivos em `/dev/hdb1` no dispositivo externo `/dev/hda9`:

```
mkfs.ext3 -O /dev/hdb1 /dev/hda9
```

Para informar ao sistema de arquivos em `/dev/hdb1` onde encontrar o journal externo, usa-se o parâmetro `-J` do comando `tune2fs`:

```
tune2fs -J device=/dev/hda9
```

Após o usuário determinar todas as opções para criar o sistema de arquivos de acordo com as suas necessidades, a opção `-n` pode ser utilizada para simular a criação do sistema de arquivos. Deste modo, o usuário vê imediatamente se as opções escolhidas são adequadas.

Existem ainda outras opções que podem ser interessantes em casos especiais. Por exemplo, uma delas reserva mais de 5% do espaço disponível na partição para o usuário root (`-m`), enquanto outra desativa funções que são ativadas por padrão (`-O`).

Outros programas importantes são o `dumpe2fs`, que mostra informações sobre o sistema de arquivos, fornecendo o conteúdo do superbloco e dos grupos de bloco, ou o `e2fsck`, que verifica a integridade de sistemas de arquivos Ext 2 e Ext 3, à procura de erros. O `e2fsck` normalmente não precisa ser utilizado com um sistema Ext 3, a única exceção é quando o superbloco está danificado e necessita de verificação. O uso do LVM (veja o artigo na página 38 desta edição) permite que se aumente ou diminua, sem perda de dados, o tamanho de uma partição com o utilitário `e2fsadm`, que controla o redimensionamento on-line e off-line. Os recursos de “dump” e “restore” do Ext 2 continuam funcionando também com o Ext 3, de modo que o usuário ou o administrador do sistema não precisa mudar seus hábitos, nem suas ferramentas.

ReiserFS: Opções de montagem

Você encontra informações detalhadas sobre o sistema de arquivos ReiserFS em

<http://www.namesys.com/mount-options.html>

conv: Com ele o ReiserFS versão 3.6 pode montar um sistema de arquivos versão 3.5, utilizando o código da versão 3.6 para os novos objetos criados. Depois disso, as ferramentas da versão 3.5 do ReiserFS não funcionam mais para este sistema de arquivos.

dontpanic (somente para reiserfs-raw):

Ignora erros de entrada e saída enquanto o journal está sendo processado. Somente disponível em modo *reiserfs-raw*.

hash=rupasov: Algoritmo de hash muito rápido para diretórios enormes e nomes de arquivos incomuns. Perigosa de usar, uma vez que colisões hash são bastante prováveis. Colisão hash é o nome dado à situação que ocorre quando dois argumentos diferentes em uma função de hash produzem resultados idênticos.

hash=tea: Altamente aleatório e com baixa probabilidade de colisões hash, mas reduz o desempenho. Deve ser utilizada caso a mensagem EHASHCOLLISION ocorra e o algoritmo de hash *r5* esteja sendo usado.

hash=r5: Hash padrão.

hash=detect: A função hash é reconhecida automaticamente quando o sistema de arquivos é montado e escrita no superbloco.

hashed_relocation: Procura ajustar o alocador de bloco. Pode melhorar a performance. Ver também *no_unhashed_relocation*.

no_unhashed_relocation: Procura ajustar o alocador de bloco. Pode melhorar a performance. Ver também *hashed_relocation*.

noborder: Desativa o *border allocator algorithm*. Pode aumentar o desempenho.

nolog: Desativa o Journaling. Pode melhorar a performance. Cuidado: isto desativa a recuperação rápida após um crash.

notail: Desativa a ferramenta *tail*. Se ocorrerem problemas com um programa durante sua operação normal (como o *lilo*, por exemplo), o ReiserFS não deverá armazenar arquivos minúsculos diretamente na árvore.

pgc=LOW, HIGH: Ativa o "coletor de lixo passivo" (Passive Garbage Collector). Só funciona em conjunto com o driver *reiserfs-raw*, que não está no kernel padrão.

raw: Monta o sistema de arquivos no modo *raw*. Não disponível no kernel padrão.

replayonly: Reverte as transações no journal sem montar o sistema de arquivos. Geralmente, somente o *fsck* utiliza esta opção.

resize=valor: Aumenta o tamanho de uma partição que está montada, em particular dispositivos LVM (Logical Volume Management). Requer a Resizer-Utility, que pode ser baixada do servidor de FTP da Namesys: ftp.namesys.com/pub/reiserfsprogs.

```
mkfs.jfs -J device=external ?
-journal device
```

O dispositivo (*device*) no final da linha de comando acima é a partição na qual o sistema de arquivos será criado (ou seja, algo como */dev/hd?n*). Num mesmo comando se pode usar ou *journal_dev* ou *external-journal*; não ambos ao mesmo tempo. Quem precisa de compatibilidade com o OS/2 deve usar a opção *-O*, para que o sistema se torne insensível a maiúsculas e minúsculas. A dimensão dos blocos é fixa em 4096 bytes. Quanto a isto o usuário nada pode fazer.

Há apenas mais três outras aplicações disponíveis para o JFS: com *jfs_tune* pode-se, entre outras coisas, listar o nome do volume e sua UUID. O comando *jfs_debugfs* executa diversas ações de baixo nível, como substituir dados em um offset com uma string hexadecimal, para fins de depuração de problemas. Por fim, *jfs_fscklog* extrai e mostra o conteúdo dos relatórios do *fsck*.

Reiser pé-de-chumbo

O ReiserFS v3.6 é, ao mesmo tempo, um sistema de arquivos estável e um ótimo brinquedo para os famintos por performance. O conjunto de programas *Reiser fsprogs* [3] contém vários utilitários com opções de ajuste e ferramentas de recuperação de erros. Por outro lado, não existem programas de dump/restore. É possível, entretanto, aumentar e diminuir o tamanho do sistema de arquivos.

Ainda que o ReiserFS seja o sistema de arquivos padrão do SuSE Linux, não há qualquer informação sobre ele disponível na distribuição. A única coisa que existe é o programa de formatação *mkfs.reiserfs*. Não há sequer uma página de manual ou qualquer sinal de documentação no diretório */usr/share/doc*. O interessante é que nem mesmo na documentação do kernel aparece alguma referência. Só resta ir à Internet e se dirigir à homepage do líder do projeto, Hans Reiser [3]. A visita ao site vale a pena: lá se encontram patches especiais para os corajosos de plantão, e os membros da lista de discussão respondem a muitas questões, ajudando a eliminar dúvidas.

Um novo sistema de arquivos é criado com a instrução *mkreiserfs*. Até existe uma opção de ajuste de tamanho de bloco, mas somente a partir da recém-

JFS: Sovina em opções

A versão do JFS para Linux atingiu um estágio que já permite a realização de testes sem dificuldades. A inadequação atual para o uso em ambientes de produção (pelo menos rodando em Linux) fica clara pela curta lista de opções. Por outro lado, os resultados do testes de desempenho aplicados ao JFS atingiram valores excelentes (ver artigo a partir da página 23). Para análises detalhadas há uma interface de depuração no código do kernel. Entretanto, no que tange a programas que possam interagir com o JFS a nível de usuário (i.e., em "user space"), o administrador de sistemas só dispõe da ferramenta *jfs_fscklog* do pacote *Jfsutils* [2], que extrai o conteúdo do arquivo de registro gerado pelo *fsck* e o exibe no console.

Comparativamente, o JFS é pobre em de opções a nível de usuário. Assim, não é possível, por exemplo, ajustar o tamanho de bloco. Entretanto, as pági-

nas de manual (*man pages*) são bem cuidadas e fornecem bons exemplos, ao contrário da documentação disponível para os outros sistemas de arquivos. Algumas opções do *mount* visam igualmente o tratamento de erros, e fornecem informações importantes para as listas de discussões.

Cadê as aplicações?

O comando *mkfs.jfs* cria um sistema de arquivos, e com o parâmetro *-c* o usuário pode procurar por blocos. Com o JFS também é possível colocar o journal em outro dispositivo (*-J*). O comando:

```
mkfs.jfs -J journal_dev ?
external-journal
```

cria o journal no dispositivo indicado. Subseqüentemente, a ligação entre o journal e o sistema de arquivos, que agora será realmente gerado, ocorria via *device*, através do comando:

lançada versão 4 o usuário pode selecionar seu tamanho. O ReiserFS seleciona outras opções sozinho, caso elas não sejam especificadas, tais como algoritmos de hash (*r5*, *rupasov*, *tea*) ou se a versão 3.5 ou a 3.6 deve ser instalada.

As opções para o journal se restringem ao tamanho do journal (-s), se ele se encontra num dispositivo externo (-j) e a partir de qual offset ele começa (-o). Além disso, o usuário pode ajustar a dimensão máxima da transação – da mesma forma que ocorre em uma transação de um banco de dados, transações de sistemas de arquivos com journaling tratam seqüências de modificações dos metadados do sistema de como uma única operação atômica, ou seja, ou ela ocorre inteiramente ou simplesmente não ocorre. O ajuste do tamanho da transação é sempre seguro, uma vez que o *mkreiserfs* o adapta automaticamente, caso o usuário use um valor incorreto. Quem utiliza o UUID (Universally Unique Identifier), pode ajustá-lo com a opção *-u*. Sem esta opção, o *mkreiserfs* escolhe o valor ideal automaticamente.

ReiserFS na oficina

Logo após a inicialização do sistema, entra em cena o programa de verificação de integridade e reparos do sistema de arquivos, *reiserfsck*. Na ocorrência de um crash, ele controla a consistência do sistema de arquivos, usando a opção padrão (*check*) e restabelece o último estado do sistema armazenado no journal. Se o *check* constata um erro grave, tal como a inconsistência de um sistema de arquivos, ele pede ao usuário para que tome as providências necessárias, e mostra no console a opção correta a ser usada. Para inconsistências pequenas, como nomes de diretórios inválidos ou truncados, ele recomenda usar a opção *fix-fixable*. Entretanto, se o sistema de arquivos foi corrompido de forma mais grave, o usuário deve tentar a opção *rebuild-tree*. O *reiserfsck* irá reconstruir toda a árvore do sistema de arquivos, mas isto leva muito tempo. E, piorando a situação, ainda há risco de, ao final da operação, o sistema de arquivos estar completamente destruído.

Se aparecer a mensagem de erro *read_super_block: cant't find a reiserfs file system*, o usuário executar o comando:

XFS: Opções de montagem

Você encontra informações detalhadas sobre o sistema de arquivos XFS em */usr/src/linux/Documentation/filesystems/xfs.txt*.

biosize=valor: Configura o tamanho preferido de I/O com buffer (o padrão é 64 kBytes). O argumento *valor* deve ser expresso como o logaritmo de base 2 do tamanho de I/O desejado.

ikeep: Mantém grupos de inodos vazios, em vez de devolvê-los ao pool de memória livre.

logbufs= valor: Define o número de buffers de registro (log) na memória central. Pode-se usar valores de 2 a 8. Pode melhorar o desempenho do sistema, mas reduz a quantidade de memória livre.

logbsize=valor: Define o tamanho dos buffers de registro (log). Pode-se usar os valores 16, 32, 64, 128 e 256 kBytes. Sistemas com mais de 32 MBytes de RAM usam 32 kBytes por padrão.

logdev=dispositivo Cria *journals* externos.

noalign: As alocações de dados não são alinhadas às bordas de "stripe units".

noatime: Não atualiza os *timestamps* durante acessos de leitura.

norecovery: Monta o sistema de arquivos sem a recuperação de registro (Log-Recovery). O sistema de arquivos deve ser montado em modo somente de leitura.

osyncisync: Atualiza os timestamps dos arquivos que tem o flag *O_SYNC* ativado. Sem esta opção a performance deve ser melhor. Entretanto, os arquivos perdem a informação de timestamp no caso de um crash.

quota, usrquota, uqnoenforce: Ativa o sistema de quotas para usuários, pode forçar que os limites sejam respeitados.

grpquota, gqnoenforce: Ativa o sistema de quotas para grupos e pode forçar que os limites sejam respeitados.

sunit=valor e swidth= valor: Estabelece o tamanho da stripe unit em intervalos de 512 bytes (unidades de bloco). A opção *swidth* deve ser configurada um múltiplo de valor de *sunit*.

Importante para dispositivos RAID: caso se modifique o layout do disco depois que o sistema de arquivos tenha sido criado, as informações podem ser sobrescritas no superbloco pelo usuário *root*.

nouuid: Não verificar via UUID se o sistema de arquivos for duplamente montado. Útil na montagem do LVM-Snapshots.3

JFS: Opções de montagem

Você encontra informações detalhadas sobre o sistema de arquivos JFS em */usr/src/linux/Documentation/filesystems/jfs.txt*.

iocharset=nome: Conjunto de caracteres que o sistema deve utilizar para converter de Unicode para ASCII.

resize=n: Aumenta o tamanho do volume online em *n* blocos.

nointegrity: Evita que o journal seja escrito. Interessante quando o backup de um volume precisa ser restaurado.

integrity: Modificações nos metadados são protocoladas pelo journal. Deve ser utilizada após restaurar um backup com a opção *nointegrity*, em conjunto com uma remontagem.

errors=continue: A montagem continua normalmente, mesmo se ocorrer um erro no sistema de arquivos.

errors=remount-ro: No caso de erro, o sistema de arquivos é novamente montado em modo somente leitura.

errors=panic: A máquina pára no caso de erro no sistema de arquivos.

```
reiserfsck -rebuild-sb
```

Neste caso, o superbloco provavelmente está corrompido, de modo que o sistema de arquivos não pode ser acessado. O usuário pode restaurar o superbloco com *rebuild-sb*, se na partição realmente houver um sistema de arquivos Reiser.

Expandindo o ReiserFS

O programa *resize_reiserfs* ainda está no estágio beta e por isso deve ser usado com cautela. A ferramenta é capaz de

aumentar ou reduzir o tamanho do sistema de arquivos, que não deve estar montado durante a operação. O comando é bastante simples de usar:

```
resize_reiserfs -s -2G /dev/hdb2
```

Com isso diminui-se o tamanho do sistema de arquivos em */dev/hdb2* em 2 GBytes. Para aumentar o sistema de arquivos em 2 GB, basta substituir *-2G* por *+2G*. Infelizmente a coisa não é assim tão simples: além de executar a

Sysctls do XFS

• *fs.xfs.stats_clear* (Min: 0; Padrão: 0; Max: 1): O valor 1 apaga as estatísticas em `/proc/fs/xfs/stat` e `/proc/fs/xfs/stat`.

• *fs.xfs.sync_interval* (Min: HZ, Padrão: 30*HZ; Max: 60*HZ): Estabelece o intervalo em que o `xfsyncd` escreve os metadados no disco.

• *fs.xfs.error_level* (Min: 0; Padrão: 3; Max: 11): Define a extensão dos relatórios de erro. Gera informações mais ou menos detalhadas e rastreia shutdowns do sistema de arquivos. Os seguintes valores de nível de erro estão definidos: OFF: 0; LOW: 1, HIGH: 5.

• *fs.xfs.panic_mask* (Min: 0; Padrão: 0; Max: 127): Opção para depuração de erros. Em casos específicos, a função `BUG()` é chamada.

• *fs.xfs.irix_symlink_mode* (Min: 0; Padrão: 0; Max: 1): Controla se os links simbólicos são criados com o modo padrão (0777) ou se o modo depende do padrão especificado com `umask` (modo IRIX).

• *fs.xfs.irix_sgid_inherit* (Min: 0; Padrão: 0;

Max: 1): Verifica os arquivos que o usuário coloca em diretórios SGID. A opção faz com que o bit ISGID seja apagado, caso o ID de grupo de arquivos novos não coincida com os IDs de grupo efetivos ou com um dos IDs adicionais dos diretórios pais.

• *fs.xfs.restrict_chown* (Min: 0; Padrão: 0; Max: 1): Verifica se usuários não privilegiados podem usar o comando `chown` para transferir a propriedade sobre arquivos a outros usuários.

• *vm.pagebufstats_clear* (Min: 0; Padrão: 0; Max: 1): O valor 1 apaga imediatamente todas as estatísticas em `/proc/fs/pagebuf/stat`.

• *vm.pagebufflush_age* (Min: 1*HZ; Padrão: 15*HZ; Max: 300*HZ): A idade a partir da qual o conteúdo dos buffers de metadados são escritos nos dispositivo de bloco.

• *vm.pagebufflush_int* (Min: HZ/2; Padrão: HZ; Max: 30*HZ): O intervalo no qual a lista de buffers de metadados é verificada.

operação anterior, o usuário ainda tem que adaptar a partição ao novo tamanho do sistema de arquivos. No caso de um aumento, isso deve ser feito antes, no de uma redução, depois. O tamanho do journal, bem como o tamanho máximo de transação, podem ser alterados com o programa `reiserfstune`. Além disso, com este comando o usuário também pode mover o journal para um outro dispositivo de bloco.

O programa `debugreiserfs` auxilia na busca por erros. Chamado sem opções, ele retorna o superbloco do dispositivo indicado. Neste contexto, a opção `-p` é interessante: com ela o `debugreiserfs` encontra e extrai os metadados do sistema de arquivos para que a equipe de desenvolvedores do ReiserFS possa eliminar o erro.

```
debugreiserfs -p /dev/hda2/ | gzip -c > hda2.gz
```

Uma desvantagem para muitos usuários de Ext 2/3 que consideram uma migração para o ReiserFS é que, até o momento, não existe ferramenta de Dump/Restore para este sistema. As FAQs do ReiserFS recomendam usar o utilitário `tar` para fazer backups incrementais. Entretanto, usuários da distribuição SuSE-Pro ganham um sistema de backups completo via YaST2.

XFS: para os exigentes

Juntamente com o Ext 3, o XFS dá a impressão de ser o sistema de arquivos mais completo. Ele não só disponibiliza numerosas opções, como também sua documentação é exemplar. É o único sistema de arquivos que permite colocar Sysctls no sistema `/proc` (ver Tabela 1). O tamanho de bloco é ajustável até 4 kB. Usuários esfomeados por performance são servidos pelo XFS com opções para diversos tamanhos de buffer. Destaques são o suporte a quotas no kernel e as ferramentas no pacote do `xfsprogs` [4]. Além disso, o pacote contém ferramentas de dump/restore próprias do XFS.

Funcionalidade total!

Nos casos mais simples, o usuário formata uma partição com o comando:

```
mkfs.xfs /dev/sda1
```

Se o registro de metadados (o journal) deve residir em outra partição, entram em cena as seguintes opções:

```
mkfs.xfs -l logdev=/dev/sdb2, size=10000b /dev/sda1
```

Com isto a primeira partição do primeiro disco SCSI é formatada, e o journal é colocado na segunda partição do segundo disco.

O XFS se comporta de maneira bastante flexível no quesito tamanho de bloco – mas infelizmente não no Linux. Blocos de 512 bytes a 64 kB são possíveis, mas no Linux o tamanho do bloco pode ser, no máximo, do tamanho do Pagesize, que é limitado a 4 KBytes em sistemas x86. O usuário pode, entretanto, selecionar um tamanho menor.

As outras opções oferecidas pelo XFS estão agrupadas de acordo com funções para manipulação da área de dados do sistema de arquivos, de inodes, do journal e para ajuste do tempo real do sistema. Cada um dos grupos fornece dados de uma determinada área do disco rígido. Assim, pode-se ajustar o tamanho da fatia de sistemas RAID, ou indicar o número de inodes variáveis em porcentagem. Uma visão geral dos parâmetros do sistema é fornecida com a opção `-N`.

O XFS também pode ser expandido: o comando `xfs_growfs` aumenta a seção dos dados para o maior tamanho possível (com a opção `-d`). A opção `-D` permite ao usuário indicar o tamanho dos blocos. A seção de tempo real do sistema de arquivos pode ser aumentada com a opção `-r`. Como no caso da seção de dados, um tamanho determinado pode ser fornecido com a opção `-R`.

Geralmente se utiliza o aumento do tamanho do sistema de arquivos em combinação com o LVM (Logical Volume Manager), simplesmente acrescentando um outro disco ao sistema, e adicionando este disco ao grupo de volumes.

Os especialistas também têm com o que se divertir: com `xfs_db` é possível manipular certos blocos, lendo-os ou escrevendo sobre eles. A opção `blockuse` fornece o nome do arquivo que utiliza um determinado bloco. A enorme lista de opções do `xfs_db` mostra que, graças ao XFS, o Linux já tem condições de concorrer de igual para igual com os sistemas UNIX tradicionais. ■

INFORMAÇÕES

- [1] Ezfsprogs: <http://ezfsprogs.sourceforge.net/ext2.html>
- [2] Jfsutils: <http://www-124.ibm.com/jfs/>
- [3] Reiserfsprogs: <http://www.namesys.com/download.html>
- [4] Xfsprogs: <http://oss.sgi.com/projects/xfs/download.html>

Sistemas de arquivos criptografados para Debian, Gentoo e Red Hat

Top Secret!

Criptografar sistemas de arquivos inteiros é também uma maneira de proteger os dados em caso de roubo do disco rígido. Enquanto os usuários de SuSE Linux podem ativar a criptografia do sistema de arquivos durante a instalação, algum trabalho manual é necessário para usuários de outras distribuições.

POR CHRISTIAN NEY



Usuários de laptop têm os seus dados à mão aonde quer que estejam. O que por um lado é prático, torna-se um problema quando o portátil é roubado. Normalmente em tais casos, os arquivos contidos no sistema caem rapidamente em mãos erradas. Mesmo aqueles que somente levam o equipamento para reparos podem cair nas mãos de um técnico curioso, que terá tempo mais do que suficiente para fuçar nos arquivos particulares.

Existem técnicas de codificação para a proteção de dados confidenciais, e elas podem ser instaladas sem muito trabalho: A GNU Privacy Guard (GPG, [1]) codifica arquivos individuais sob demanda. Porém, para proteger vários arquivos, este procedimento é muito desconfortável. Em tais casos, é melhor codificar todo o sistema de arquivos de uma vez. A distribuição SuSE já mostra há anos (ver Figura 1) que isso pode ser feito de forma amigável. O sistema de arquivos criptografado pode ser criado durante a instalação, caso desejado. Em outras distribuições, isso pode ser feito com algum trabalho manual – e mesmo melhorado em alguns pontos. A distribuição Gentoo facilita muito a vida do usuário neste particular.

Casa trancada

Tomamos como exemplo para o nosso artigo a criptografia do sistema de arquivos instalado e montado separadamente em `/home`. Ele deve ser montado durante

a inicialização do sistema, independente de estar criptografado, e a chave de criptografia deve ficar em um chaveiro USB. Esta chave substitui a senha comum, servindo de base para a codificação do sistema de arquivos durante a inicialização do dispositivo *cryptloop*. A vantagem deste método é que a chave consiste de valores aleatórios e é bem mais longa que a maioria das senhas. Assim, os chamados “ataques de dicionário” são excluídos automaticamente: um agressor teria que realmente tentar todas as possibilidades (força bruta) para descobrir qual é a chave de codificação. É quase impossível que ele obtenha sucesso com uma chave de tamanho razoável.

No Linux há diversas possibilidades de criptografia de sistemas de arquivos inteiros. A criptografia via dispositivo *loopback*, que se serve das funções criptográficas do kernel [2], oferece a maioria dos algoritmos disponíveis e, além disso, é muito simples de usar quando comparada com outras soluções.

Sanduíche

Os dispositivos de loop utilizam uma camada intermediária, que se situa entre o dispositivo de bloco (por exemplo `/dev/hda1`) e o sistema de arquivos nela contido. Normalmente o sistema de arquivos acessa o dispositivo de bloco diretamente, como mostrado na Figura 2.

Dispositivos de loop comportam-se como dispositivos de bloco comuns. Entretanto, eles retransmitem os dados

para um dispositivo normal ou para um arquivo (Figura 2). Qualquer um que já tenha montado uma imagem ISO com a opção *loop* conhece este procedimento.

Os dispositivos de loopback também são capazes de modificar os dados durante a operação. Com o dispositivo *cryptloop*, o kernel intervém a cada escrita e leitura para codificar e decodificar os dados (Figura 2). O usuário não percebe nada – com exceção do pedido inicial de uma senha. Mesmo o pedido de senha pode ser evitado, se a chave for alojada num chaveiro USB. Infelizmente os patches necessários para criar sistemas de arquivos criptografados no Linux não estão contidos nos kernels de todas as distribuições Linux por padrão.

- Para o Debian Sarge é necessário recompilar o kernel juntamente com o pacote *kernel-patch-cryptloop*.
- A partir de sua versão 8, o Red Hat (agora Fedora) já dispõe de suporte a sistemas de arquivos com *cryptloop*. Contudo, somente o módulo AES está disponível por padrão.
- O kernel padrão do Gentoo, *gentoo-sources* já contém o patch necessário há muito tempo. Porém, é preciso que as opções necessárias sejam selecionadas antes da compilação (Figura 3).

Pode-se verificar facilmente se a distribuição fornece os patches necessários: produza um arquivo codificado e tente integrá-lo a um sistema de arquivos nor-

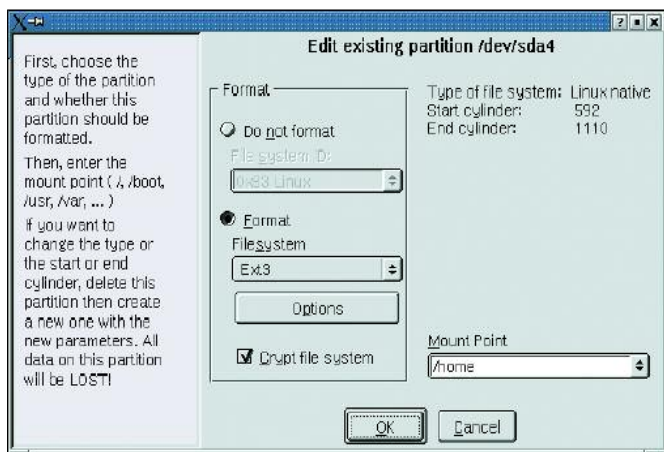


Figura1: No SuSE Linux, basta um clique do mouse na janela correta do YaST para ter o sistema de arquivos `/home` codificado.

mal. É preciso antes fornecer, com a ajuda do comando `dd`, um arquivo que mais tarde irá conter o sistema de arquivos codificado. Um tamanho de 20 MBytes para este arquivo é adequado para o teste.

O dispositivo de loop

O comando `losetup` produz um dispositivo de loop a partir do arquivo criado. Ele pede para isso uma senha de codificação. Para fins de teste uma senha é suficiente; mais tarde criaremos uma chave aleatória. Para inserir a senha é preciso cautela: o `losetup` só pede a senha uma única vez. Quem errar a digitação perde o acesso aos dados codificados. Para finalizar, deve-se ainda criar um sistema de arquivos no dispositivo de loop, montar o sistema de arquivos e colocar nele alguns arquivos, conforme mostrado abaixo:

```
dd if=/dev/zero of=/root/cryptfile 2
bs=1024k count=20
losetup -e aes -k 128 /dev/loop0 2
/root/cryptfile
mkfs -t ext2 -m 0 /dev/loop0
mount -t ext2 /dev/loop0 /mnt
cp /bin/{mv,rm} /mnt
```

Se o `losetup` retornar uma mensagem de erro logo de início, provavelmente o kernel ainda não está em condições de lidar com sistemas de arquivos criptografados. A Crypto-API está faltando. A referência [3] esclarece como adicioná-la ao kernel. Não é necessário recompilar todo o kernel – a CryptoAPI é composta apenas de módulos que o kernel carrega durante a execução.

Também é necessário adicionar suporte a chaves USB no kernel. As opções apropriadas para isso são *USB-Mass-Storage*, *SCSI-Disk-Support* e *SCSI emulation support*. A maioria dos kernels que vêm com as distribuições atuais, normalmente já contém os módulos apropriados.

Ferramentas do usuário

Além do suporte à codificação e decodificação no kernel, são necessários programas no espaço do usuário para criptografar os sistemas de arquivos e depois montá-los. Para isso o `losetup` e o `mount` (presente no pacote `util-linux`) precisam ser instalados e configurados.

- O pacote `util-Linux` da Debian Sarge já é preparado para as condições de sistemas de arquivos codificados, não sendo necessário um patch.
- Até a versão 9 do Red Hat, o pacote `util-linux` não continha os patches necessários. Neste caso também é necessário trabalho manual. As etapas são muito bem explicadas em [4].
- O Gentoo acrescenta ao pacote `util-linux` as funções para tratar dos sistemas de arquivos criptografados com a opção (“use flag”) `+crypt`.

Antes de colocar as mãos na massa, chegou a hora de fazer aquele backup que você estava planejando fazer tempo. Caso contrário, uma vez que nossas instruções a seguir levarão à exclusão de todo o sistema de arquivos `/home` origi-

nal, os dados contidos nele seriam irremediavelmente perdidos.

Um sistema de arquivos codificado só representa um ganho genuíno de segurança quanto não restar no disco nenhum resquício da velha partição. Isso significa destruir totalmente os dados do disco: exclusão ou formatação não são suficientes. Certifique-se de que os arquivos originais estão no backup, eles serão, mais tarde, recolocados no novo sistema de arquivos criptografado.

Para inutilizar o antigo sistema de arquivos, é mais simples enchê-lo com valores aleatórios: o administrador desmonta o sistema de arquivos `/home` e usa o gerador de números aleatórios do kernel para, literalmente, entupir a partição com “lixo”. No exemplo a seguir, a partição `/home` fica em `/dev/hda6`. Quem dá valor à melhor eficiência na geração de valores aleatórios e dispõe de bastante tempo, deve dar preferência ao `/dev/random`, ao invés de usar o `/dev/urandom` (ver Quadro 1: “Eficiência demanda tempo”).

```
dd if=/dev/urandom of=/dev/hda6 2
bs=4k conv=notrunc
```

Este comando sobrescreve todo o dispositivo de bloco com valores aleatórios. Desta forma, fica mais difícil depois diferenciar “lixo” e dados codificados. A sobrescrita leva mais ou menos tempo, dependendo da quantidade de entropia existente no gerador de números aleatórios e do tamanho do sistema de arquivos. Por incrível que pareça, mesmo todos estes procedimentos ainda não são suficientes para impedir restauradores profissionais de dados de recuperar os dados antigos. Ainda há magnetismo remanescente suficiente no disco rígido [5]. Quem for realmente paranóico deve repetir este procedimento várias vezes.

Quadro 1: Eficiência demanda tempo.

Ambos os dispositivos `/dev/random` e `/dev/urandom` são interfaces para o gerador de números aleatórios do kernel. Eles coletam eventos os mais aleatórios possíveis e os reúnem em uma “pilha de entropia”. Cada acesso ao `/dev/random` reduz a quantidade de entropia existente na pilha.

Caso não haja mais entropia suficiente, o kernel bloqueia o processo de leitura até que colete novamente a quantidade necessária de valores randômicos. Isso confere alta qualidade ao gerador de valores aleatórios. Muitas funções criptográficas (por exemplo: OpenSSL, GPG e outros) usam este gerador.

O dispositivo `/dev/urandom`, entretanto, retorna todo o volume de dados requisitado, sem examinar se há entropia suficiente. Com isso pode acontecer que agressores consigam tirar conclusões sobre os dados aleatórios retornados. Contudo, tais valores são suficientes para tarefas simples.

A chave do sistema de arquivos

Como próximo passo, vamos gerar o arquivo com a chave. Esta chave serve para codificar e decodificar os dados do sistema de arquivos criptografado. Para evitar que a chave possa ser quebrada por um ataque de dicionário, ela deve ser a mais longa e aleatória possível. Somente assim haverá realmente uma vantagem em comparação com uma senha comum, geralmente mais curta. Já que o arquivo com chave de codificação deve estar no chaveiro USB, ele deve estar montado no sistema, o que faremos a seguir em `/mnt/usb/`:

```
head -c 2880 /dev/random | ➤  
uuencode -m - | head -n 65 | tail ➤  
-n 64 | gpg --symmetric -a > ➤  
/mnt/usb/key.gpg
```

A linha de comando acima lê somente 2880 bytes do `/dev/random` e utiliza então `uuencode` para codificar os valores aleatórios em Base64. Os comandos `head` e `tail` eliminam as saídas de texto puro do `uuencode`.

O resultado já tem características aleatórias muito boas, e a codificação final com GPG melhora ainda mais a qualidade. Para isso o GPG requer uma frase-senha, que serve de chave simétrica. Apenas com ela é possível decodificar os dados – o que significa dizer que, se este arquivo-chave for perdido, os dados ficarão inacessíveis. Por esta razão, é extremamente importante que uma cópia dele seja mantida em lugar seguro.

A codificação adicional com GPG também serve para proteger a chave contra usuários não autorizados: ao invés de

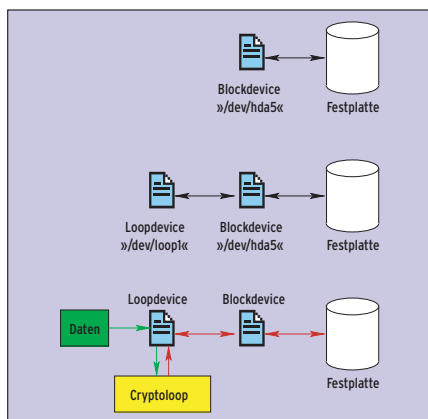


Figura 2: O kernel dá acesso a um dispositivo de bloco no disco rígido (acima). O dispositivo de loop (centro) entra antes do dispositivo de bloco. Com a criptografia via loopback (abaixo) somente dados criptografados vão parar no disco rígido.

utilizar a versão codificada, pode-se decodificar novamente o arquivo-chave com auxílio da frase-senha do GPG e usar os dados resultantes como chave. Deste modo, um agressor precisaria não só estar em poder do chaveiro USB, como também conhecer a frase-senha do GPG para conseguir acesso aos dados.

Os preparativos agora acabaram: o próximo passo é configurar o dispositivo de loop. Ao mesmo tempo, são oportunas algumas reflexões sobre o algoritmo de codificação a ser utilizado e comprimento de chave. O exemplo a seguir usa o Twofish [6]. Este algoritmo é empregado há alguns anos em muitos produtos e foi avaliado por muitos especialistas em criptografia. Ele é considerado bastante maduro (veja o Quadro 2).

```
losetup -e twofish -k 128 ➤  
/dev/loop0 /dev/hda6 -p 0 < ➤  
/mnt/usb/key.gpg
```

Quadro 2: “Decifra-me, ou te devoro!”

Um bom ponto de referência para a seleção de um algoritmo apropriado é oferecido pelo comparativo [8] da AES (Advanced Encryption Standard), escrito, entre outros, pelo papa da criptografia Bruce Schneier. Ele mostra que o Rijndael e o Twofish são os algoritmos mais rápidos e também os mais freqüentemente utilizados. No entanto, o Rijndael perde em velocidade na medida em que o comprimento da chave aumenta.

O Twofish oferece um pouco mais de segurança [9]. Apesar de o algoritmo de Rijndael ser um pouco mais recente, já existem ataques teóricos contra este algoritmo. De qualquer maneira, o nível de segurança oferecido por Rijndael, considerado o melhor segundo a AES, também deve ser totalmente suficiente.

Outro ponto é a escolha de um comprimento adequado para a chave. Pode-se escolher 128, 192 ou 256 bits. Quanto maior o comprimento, mais seguro é o algoritmo contra ataques de força bruta; por outro lado, a velocidade diminui. Se houver deficiências na codificação usada para gerar a chave, ou seja, se pudermos de algum modo inferir os dados contidos nela, de pouco adianta o seu maior comprimento [10]. Por esta razão 128 ou 192 bits são considerados suficientes.



Figura 3: Os kernels padrão (sem patches suplementares) já contém opções de criptografia. Para ativá-las basta escolher *Cryptographic API* e os algoritmos desejados sob o menu *Cryptographic Options*, usando uma das muitas interfaces de configuração kernel.

Listagem 1: Script de inicialização

```
01 #!/bin/sh -e
02
03 case "$1" in
04   start)
05     /sbin/modprobe sd_mod
06     /sbin/modprobe usb-storage
07     /sbin/modprobe cryptoapi
08     /sbin/modprobe cryptoloop
09     /sbin/modprobe twofish
10     /bin/mount -t vfat >
11     /dev/sda1 /mnt/usb
12     /sbin/losetup -e twofish >
13     -k 128 /dev/loop0 /home -p 0 >
14     < /mnt/usb/key.gpg
15     /bin/mount /dev/loop0 /home
16     /bin/umount /dev/sda1
17     ;;
18   stop)
19     /sbin/losetup -d /dev/loop0
20     /bin/umount /dev/loop0
21     ;;
22   esac
```

Este comando inicializa o dispositivo de loop `/dev/loop0` com o algoritmo Twofish. O comprimento da chave é de 128 bits. O dispositivo de loop armazena os dados codificados no dispositivo de bloco `/dev/hda6`. O programa espera por uma senha como base para a chave, neste caso, não vem pelo teclado, e sim do arquivo `key.gpg` no “chaveiro”, redirecionado através da entrada padrão com o comando: `-p 0 < /mnt/usb/key.gpg`.

Aquecendo os motores...

Agora o dispositivo de loop está pronto para ser usado com qualquer sistema de arquivos. Sistemas de arquivos com journaling podem, contudo, em casos desfavoráveis, causar problemas: eventualmente o dispositivo de loop escreve no disco numa seqüência diferente da esperada pelo sistema de arquivos. O journal torna-se, então, totalmente inútil e o que pode levar a inconsistências graves no sistema de arquivos no caso de crashes.

```
mkfs -t ext2 /dev/loop0 mount >
/dev/loop0 /mnt
```

No nosso exemplo utilizamos como sistema de arquivos o Ext 2. Após a montagem, é aconselhável criar alguns arquivos para testar o novo sistema de arquivos. Depois, desmontar o sistema de arquivos novamente e desativar o dispositivo de loop:

```
umount /mnt
losetup -d /dev/loop0
```

Para evitar surpresas desagradáveis mais tarde, deve-se repetir diversas vezes a criação do dispositivo de loop e a montagem do sistema de arquivos. Não havendo ainda nenhum dado importante no sistema de arquivos criptografado, em caso de dúvida, pode-se começar tudo do início novamente.

Aos seus lugares...

Ainda são necessárias algumas mudanças no sistema, uma vez que todo o procedimento, inclusive a montagem do chaveiro USB, deve ocorrer automaticamente. Algumas distribuições já estão preparadas para isso. A Gentoo, por exemplo, fornece o script de inicialização `cryptoloop`. Os comentários contidos neste script esclarecem eventuais dúvidas sobre seu uso. As soluções genéricas a seguir não são tão elegantes, mas devem funcionar na maioria das distribuições comuns. A variante mais simples usa um script de inicialização já existente; geralmente `/etc/init.d/rc.local` é uma boa escolha. Mais elegante ainda seria criar um script de inicialização próprio, obviamente observando a seqüência de chamada de tais scripts pelo sistema durante o boot.

Um exemplo de `rc.local` para as versões de kernel a partir do 2.4.22 pode ser vista na Listagem 1. Os nomes dos módulos de algoritmo foram modificados a partir do kernel 2.4.22. Para as versões mais antigas os nomes devem começar com um `cipher-`. A linha 9, para um kernel mais antigo, ficaria assim: `/sbin/modprobe cipher-twofish`.

A montagem do chaveiro USB seria ainda mais confortável com a ajuda do montador automático de volumes `supermount` [7]. Para isso é necessário um outro módulo de kernel, que pode ser facilmente instalado. Adicionalmente, é necessária uma entrada especial no arquivo `/etc/fstab`:

```
/mnt/usb/mnt/usb supermount >
sync,rw,fs=auto,dev=/dev/sda1 0 0
```

Com isso, podemos economizar as montagens explícitas no script de inicialização. O `supermount` monta o chaveiro USB sob demanda. Pode-se utilizar a mesma técnica para montar automaticamente os CD-ROMs, bastando para isso alterar o dispositivo e o ponto de montagem indicados acima, com `/dev/cdrom` e `/mnt/cdrom`, respectivamente.

Segurança total!

Todos os nossos esforços terão sido em vão caso dados confidenciais ainda puderem ser encontrados sem codificação em outro lugar do disco. O pior caso é aquele em que o kernel move exatamente a zona da memória em que a chave se encontra para a partição `swap`. Outro risco são os famosos “core dumps”, que às vezes escrevem a zona memória de um processo após um erro grave no disco rígido.

Ambos os problemas são fáceis de resolver. Na área de `swap` (memória virtual) não há nenhum dado que, em condições normais, ainda seja necessário após uma reinicialização do sistema. Por isso é perfeitamente possível criptografar esta área de modo transparente, e melhor ainda, com uma chave aleatória que ninguém conheça.

A desvantagem deste método é um atraso mínimo adicional ao durante o “boot”: uma vez que o sistema codifica a partição `swap` após cada inicialização do sistema com uma outra chave, ele deve reinicializar esta partição a cada vez.

Para isso é necessário modificar o script de inicialização responsável pela montagem da partição *swap*. Com o comando `grep swapon /etc/init.d/*` identifica-se qual o script a ser modificado. Neste script, antes da linha que contém o comando *swapon*, devem ser inseridas duas linhas suplementares:

```
/bin/dd if=/dev/urandom bs=1 \
count=16 | /sbin/losetup -e \
twofish -k 128 -p 0 /dev/loop1 \
/dev/hda2
/sbin/mkswap /dev/loop1
/sbin/swapon -a
```

Estes comandos produzem um outro dispositivo de loop. Sua senha se compõe de 16 bits de valor aleatório. Como esta chave muda a cada nova inicialização, a partição *swap* é reinicializada a cada novo boot através do comando *mkswap*.

Além disso, deve-se substituir no arquivo */etc/fstab* a linha que corresponde ao dispositivo de bloco da partição *swap* (em nosso exemplo, */dev/hda2*) pelo dispositivo de loop correspondente (aqui */dev/loop1*). Os mais paranóicos devem desativar momentaneamente a partição *swap* com *swapoff -a*, sobrescrevê-la com o comando *dd* e o dispositivo */dev/urandom*, e executar manualmente os três passos descritos acima.

Adeus, "core dump"

Em quase todas as distribuições, a geração de core dumps é desativada por padrão. Tal comportamento pode ser verificado através do comando `ulimit -a | grep core`. Ele determina o número máximo de blocos que um arquivo "core" pode ocupar. Se o valor retornado pelo comando indicado for 0, o sistema não produz core dumps. Se o valor for maior que 0, é possível desativá-lo glo-

balmente no arquivo */etc/security/limits.conf*. A entrada conveniente para isso é:

```
* soft core 0
```

Na maioria dos casos, esta linha já está até presente no arquivo, mas comentada, bastando apenas remover o caractere # para ativá-la.

E o Kernel 2.6?

Quem quiser usar partições criptografadas sob o kernel 2.6, vai seguir passos semelhantes aos descritos aqui, mas deve fazê-lo apenas quando estiver usando exclusivamente o novo kernel, pois os dispositivos *cryptoloop* criados sob o kernel 2.4 infelizmente não podem ser montados pelo kernel 2.6.

Até mesmo uma atualização de kernel dentro da série 2.4 pode resultar no fato do novo kernel não mais decodificar os dados criptografados. As situações mais críticas ocorrem quando com atualizações para versões do kernel a partir da 2.4.22. Depois desta versão o kernel oficial também contém *crypto code* e os patches descritos neste artigo não são mais necessários. Infelizmente as mudanças não se restringem somente aos nomes de cada módulo (como por exemplo *aes* ao invés de *cipher-aes*), mas são tão profundas que o antigo sistema de arquivos criptografado não pode mais ser lido pela nova versão. Felizmente, a atualização do kernel 2.4.22 para outros da série 2.4 não traz mais nenhum problema no que tange a dispositivos criptografados.

Devido à flexibilidade do procedimento *cryptoloop*, há espaço para diversas variações e melhorias, que tornem o sistema ainda mais personalizável ou seguro. Quem, por exemplo, precisa de um sistema de arquivos FAT no chaveiro USB para armazenar outros dados que

possam ser lidos no Windows, pode adicionalmente criar nele um pequeno sistema de arquivos Ext 2 para guardar o arquivo-chave, de forma que ele fique protegido de modificações inadvertidas.

Na medida

Apesar das operações adicionais de codificação e decodificação, é difícil perceber perdas de desempenho nos procedimentos de leitura e escrita em laptops comuns, exceto pelo uso ligeiramente maior de CPU, quando comparamos com a velocidade de tais procedimentos em sistema de arquivos não codificado. ■

SOBRE O AUTOR

Christian Ney administra de sistemas Unix e Firewalls para uma pequena empresa aérea, e trabalha em vários projetos Open Source no tempo livre.



INFORMAÇÕES

- [1] GPG: <http://www.gnupg.org/>
- [2] CryptoAPI: <http://www.kernel.org/index.shtml>
- [3] Instalação da Crypto-API: <http://www.kernel.org/howto/nodex.php>
- [4] Patches do *util-linux* para o Red Hat/Fedora: <http://www.q-vadis.net/index.php?mID=stories&lng=en&art=5>
- [5] Peter Gutmann, "Exclusão segura de dados de meios magnéticos e de estado sólido": http://www.cs.auckland.ac.nz/~pgut001/pubs/secure_del.html
- [6] Documentação do Twofish: <http://www.schneier.com/paper-twofish-paper.pdf>
- [7] Supermount: <http://supermount-ng.sourceforge.net/>
- [8] Comparativo da AES: <http://www.schneier.com/paper-aes-performance.pdf>
- [9] Comparações entre o Twofish e o Rijndael: <http://www.schneier.com/paper-twofish-final.pdf>
- [10] Tamanho de chaves: <http://mitglied.lycos.de/cthoeing/crypto/keylen.htm>
- [11] Loop-AES: <http://loop-aes.sourceforge.net/>
- [12] Loop-AES-Readme: <http://loop-aes.sourceforge.net/loop-aes.README>

Uma alternativa: Loop-AES

O Loop-AES [11] foi desenvolvido especialmente para a codificação de sistemas de arquivos. Ele também é capaz de codificar a partição *swap* sem a necessidade de adaptações especiais. O Loop-AES usa o algoritmo AES, também conhecido como Rijndael, mas também pode usar outros algoritmos como o Blowfish, o Twofish ou o Serpent, se desejado.

Como no caso da Crypto-API, com o Loop-AES também não é necessário compilar outro kernel; as funções necessárias podem ser incorporadas ao sistema na forma de módulos. Na maioria das distribuições, entretanto, o pacote *util-linux* ainda vai precisar de um patch. Ambos estão muito bem descritos em [12].

O Loop-AES simplifica o projeto em alguns pontos, apesar da flexibilidade oferecida pela solução oferecida pela Crypto-API. Quando o administrador do sistema desejar permitir o acesso ao sistema por diferentes usuários via chave GPG, deverá dar uma olhada na solução com Loop-AES.

Aumente a taxa de transferência de dados de seus discos IDE com o hdparm



Pé na tábuca!

O desempenho de discos rígidos em sistemas Linux recém-instalados fica normalmente aquém das possibilidades oferecidas pelo hardware. O `hdparm` solta o “freio-de-mão” puxado pelas distribuições na instalação padrão.

POR THOMAS WÖLFER

A baixa velocidade na transferência de dados - mesmo com os mais modernos discos rígidos disponíveis no mercado - é o preço que atualmente se paga por uma instalação padrão de distribuições Linux como o Debian, SuSE e outros, já que elas são extremamente conservadoras na configuração de hardware, de olho na compatibilidade máxima durante o processo de instalação. O utilitário `hdparm` solta esse “freio de mão” artificial de modo simples e eficiente. Este artigo vai lhe mostrar como utilizar, em doses homeopáticas e de forma segura, o `hdparm` para tirar o máximo que seu disco rígido é capaz de oferecer. Dependendo do hardware, isso pode significar um ganho de desempenho até dez vezes maior.

O mapa da mina

Quase todos os discos rígidos atualmente no mercado oferecem suporte a diversos modos de operação, como PIO (*Program-*

med Input and Output - Entrada e Saída Programada) e UltraDMA (*Direct Memory Access - Acesso Direto à Memória*), entre outras tantas funções e características especiais; nenhuma delas é, entretanto, considerada pelas rotinas de instalação padrão de várias distros. O programa `hdparm` permite que modifiquemos a configuração do disco rígido, possibilitando que obtenhamos dele o máximo em performance.

Mas cuidado: otimizar o desempenho do HD pode ser um pouco arriscado. É imprescindível fazer um backup dos dados em disco antes de começar a realizar experimentos. Além disso, recomenda-se realizar uma fase de testes antes de proceder à configuração propriamente dita. Para isso, inicialize o computador em “Single User Mode”.

Utilizamos em nossos testes um AMD AthlonXP 2600+ com 256 MBytes de RAM e um disco rígido EIDE. Segundo as especificações do fabricante, o disco

rígido utilizado oferece uma taxa de transferência de dados máxima de 66 MByte/s. Para saber qual é a taxa de transferência atual do disco rígido, utiliza-se as opções `-T` e `-t`. O utilitário `df` pode ser usado para mostrar quais unidades de disco estão instaladas em seu sistema. Use as opções `-T` e `-t` sempre juntas. `hdparm` mede a

Mostrando as possíveis configurações

Testes repetidos com o sistema exemplo mostraram que o `hdparm` calculou valores quase idênticos para as taxas de transferência de dados, com uma pequena variação. A surpresa foi que ao invés dos 65 Mbyte/s de desempenho que deveriam ser atingidos pelo disco rígido, de acordo com o fabricante, obtivemos apenas 3,9 MByte/s. Ou seja, sem nenhuma otimização, o desempenho do disco rígido é aproximadamente 17 vezes menor do que aquilo que ele realmente pode oferecer.

Para otimizar a performance do disco rígido, vamos primeiramente verificar qual é a sua configuração atual. Para tanto, basta executar o `hdparm` sem opções, indicando apenas o dispositivo a ser analisado (Figura 2). A maioria das opções está desativada e, além disso, o kernel trata as controladoras de disco como dispositivos de entrada e saída de 16 bits (“I/O Support”). Não é de se admirar que em uma configuração como essa a taxa de transferência não passe de

meros 4 MByte/s.

```

jrtstest1:~ # hdparm -Tt /dev/hda

/dev/hda:
Timing buffer-cache reads: 1164 MB in 2.00 seconds = 581.42 MB/sec
Timing buffered disk reads: 12 MB in 3.21 seconds = 3.73 MB/sec
jrtstest1:~ #
  
```

Figura 1: Desempenho miserável, pois a configuração do disco rígido não está otimizada.

```

jrtstest1:~ # hdparm /dev/hda

/dev/hda:
multcount      = 0 (off)
IO_support     = 0 (default 16-bit)
unmaskirq     = 0 (off)
using_dma     = 0 (off)
keepsettings  = 0 (off)
readonly      = 0 (off)
readahead     = 8 (on)
geometry      = 7473/255/63, sectors = 120060864, start = 0
jrtstest1:~ #
  
```

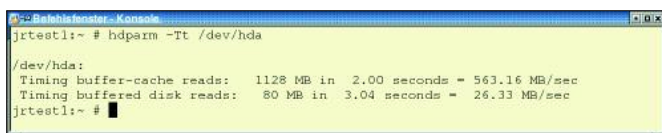
Figura 2: Executar o `hdparm` sem opções fornece uma lista que, além dos parâmetros físicos (“geometry”) do disco, mostra também se ele se encontra em modo DMA.

Dobrando o desempenho com valores seguros

Todo mundo quer melhorar o desempenho do sistema. Não é possível, entretanto, dominar todas as opções do *hdparm* logo de início. Desse modo, vamos primeiro dar uma boa olhada nas possibilidades que o comando oferece. A palavra *multcount* é uma abreviação para “Multiple Sector Count”, que especifica quantos setores são lidos em uma única interrupção de I/O. A maioria dos discos rígidos oferecem suporte para 2, 4, 8 ou 16 setores. É possível que as cabeças de leitura dos discos leiam até mais que 16 setores. A quantidade de setores que um disco rígido pode ler pode ser verificada com a opção *-i* (*Max-MultSect*). Entretanto, não se deve confiar cegamente nos valores fornecidos pelo *hdparm*: alguns discos simplesmente indicam poder suportar a leitura de um número de setores que, na prática, supera a sua capacidade real. O resultado é a perda certa de dados! Em alguns casos mais graves até o sistema de arquivos é corrompido - o manual do *hdparm*, inclusive, traz um alerta a esse respeito. O ganho de desempenho tende a ser muito grande, mas o risco de perda de dados é diretamente proporcional a ele. A melhor proteção é fazer primeiro um backup do sistema e somente depois realizar testes para se chegar ao valor ideal de *multcount*, que pode ser fixado com a opção *-m*. Por fim, a utilização de suporte a I/O indica ao *hdparm* como os dados que vêm do “bus” PCI são manipulados pela controladora de disco. Controladoras modernas não precisam rodar em modo 16 bits. O suporte a I/O pode ser configurado com a opção *-c*, conforme segue:

```
hdparm -c1 -m16 /dev/hda
```

Com esse comando o suporte a I/O é configurado em modo 1, que é um dentre vários modos de 32 bits. Todos os modos estão listados no manual do comando sob a opção *-c*. A opção *-m16* configura o sistema para leitura de 16



```

jrttest1:~ # hdparm -Tt /dev/hda
/dev/hda:
Timing buffer-cache reads: 1128 MB in 2.00 seconds = 563.16 MB/sec
Timing buffered disk reads: 80 MB in 3.04 seconds = 26.33 MB/sec
jrttest1:~ #

```

Acima de tudo o uso do modo UltraDMA acelera em muito a transferência de dados.

setores por interrupção de I/O, o que leva a resultados muito melhores na próxima bateria de testes

Já conseguimos praticamente dobrar a taxa de transferência de dados do disco rígido: de 3,9 MByte/s chegamos a 7,6 MByte/s. No entanto, ainda estamos bem longe da performance prometida pelo fabricante, que é de 66 MByte/s. Vamos ver o que é possível fazer para conseguir melhorar este cenário.

Quadruplicando a taxa de transferência com DMA

Como próxima medida de otimização vamos permitir o não-mascaramento de interrupções (*umaskirq*). Além disso, vamos ativar o modo de acesso direto à memória (DMA: *using_dma*). Normalmente o Linux entra em modo “IDLE” enquanto espera por uma interrupção - ou seja, ele não faz nada. É óbvio que com isso ele desperdiça tempo de transferência de dados, caso ele não esteja ocupado com outras tarefas. E via de

regra esse comportamento faz sentido, visto que isso contribui para aumentar a estabilidade do sistema. Contudo, se desativarmos esta proteção, aumentamos a responsividade do sistema. O problema é que, na maioria dos casos, ele se torna tão instável que nem dá mais para usar. De qualquer modo, realizar alguns experimentos com o não-mascaramento de interrupções não envolve riscos. O mesmo vale para o uso do modo DMA: temos opções para “brincar” à vontade. Mas o sistema tende a travar quando insistimos em tentar ativar as opções erradas de acesso direto à memória. Felizmente, basta reiniciar o computador para que tudo volte ao normal, de modo que podemos realizar os testes sem medo de seqüelas ou perda de dados.


Caso a documentação do disco rígido ou da controladora estejam à mão, use os modos de operação de acesso direto à memória indicados nela. Sem a documentação, será necessário realizar experimentos com o sistema (e

testá-lo) até que a configuração ideal seja encontrada. Tivemos que nos esforçar bastante até encontrar a configuração ideal de nosso sistema de testes, mostrada a seguir:

```
hdparm -d1 -X66 -u1 -m16 -cd 2 /dev/hda
```

A opção *-d1* ativa o “flag” de utilização de acesso direto à memória (*using_dma*). Além disso, também é necessário indicar qual modo de acesso deve ser utilizado. Em nosso exemplo usamos UltraDMA modo 2, configurado através da opção *-X66*. A opção *-u1* desativa o mascaramento de outras interrupções (*umaskirq*). Os resultados alcançados são surpreendentes.

A taxa de transferência de dados quintuplica com o uso dos novos parâmetros, se a compararmos com a configuração padrão. Entretanto, a nossa nova configuração não irá resistir à próxima reinicialização do sistema.



```

jrttest1:~ # hdparm -Tt /dev/hda
/dev/hda:
Timing buffer-cache reads: 1172 MB in 2.00 seconds = 584.54 MB/sec
Timing buffered disk reads: 24 MB in 3.15 seconds = 7.63 MB/sec
jrttest1:~ #

```

Figura 3: Basta configurar a controladora para realizar leituras de 32 setores por interrupção de I/O, ao invés de 16, para dobrar a taxa de transferência de dados.

Salvando a nova configuração

Como mencionado anteriormente, a configuração só permanece válida até a próxima reinicialização do sistema. Para salvá-la, precisamos adicionar o comando *hdparm* mostrado anteriormente a um dos scripts de inicialização do sistema, *rc**. Entretanto, apenas isso não basta: em determinadas circunstâncias, a controladora de disco realiza um chamado “IDE reset”. Quando isso ocorre, perde-se a configuração atual. A opção *-k1* evita que tal reset seja efetuado. Mas ela só deve ser usada após nos certificarmos de que todas as opções passadas para o *hdparm* funcionam sem problemas - ou seja, quando podemos ler e escrever dados do disco sem que nenhum alerta (“warning”) apareça no arquivo */var/log/messages*.

INFORMAÇÕES

[1] Homepage do *hdparm*: <http://www.ibiblio.org/pub/Linux/system/hardware/>

Combinando Software-Raid e Logical Volume Manager

Jogue seus dados



A pesar do aumento constante da capacidade dos discos rígidos, o problema do armazenamento de dados continua a ser uma questão complicada. Além disso, o aumento da quantidade de dados a armazenar também cresce a cada dia. Em sistemas distribuídos, especialmente, nos quais os dados das aplicações e dos usuários são armazenados em diferentes discos e partições, é muito freqüente haver problemas de espaço. Adicionar novos discos rígidos ao sistema não é realmente uma solução para o problema: em busca de espaço, não é incomum que dados acabem por ser freqüentemente transferidos de um lugar para outro.

Uma solução viável são os chamados sistemas Raid, que concatenam diversos discos rígidos em um único dispositivo lógico. O artigo [1] descreve como reconfigurar um sistema Linux para utilizar Raid5 via software sem que haja necessidade de reinstalação do sistema. A utilização de sistemas Raid, entretanto, traz consigo uma desvantagem: o sistema fica limitado ao tamanho do Raid. Dessa forma, colocar as áreas de usuários em outros volumes ou adicionar discos rígidos com maior capacidade de armazenamento exigiria necessariamente um sistema Raid ou discos rígidos extra isolados (i.e., sem redundância), o que leva de novo ao mencionado problema de distribuição de dados.

Com o Logical Volume Manager (LVM) o administrador ganha a flexibilidade que faltava para, por exemplo, aumentar

Quando o espaço no disco rígido acaba, o sonho de qualquer usuário ou administrador de sistemas é adicionar outro disco e continuar a trabalhar. O Logical Volume Manager (LVM) torna este sonho realidade: ele concatena diversos dispositivos de bloco em um único dispositivo de armazenamento homogêneo, que pode ser facilmente modificado.

POR CARSTEN WIESE

a capacidade total de armazenagem ou remover volumes mesmo com o sistema em operação. O que o LVM tem de tão especial é exatamente a capacidade de aumentar ou diminuir a quantidade de memória disponível com o sistema funcionando - desde que o sistema de arquivos também o permita.

Raid por software e Logical Volume Manager formam assim uma excelente combinação de alta disponibilidade e escalabilidade. E como “quitute” adicional temos a função de Snapshot do LVM, da qual falaremos mais adiante.

Muitas alternativas, difícil decisão

Atualmente existem duas excelentes implementações de gerenciador de volumes no Linux: de um lado temos o Logical Volume Manager (LVM), desenvolvido desde 1997 por Heinz Mauelshagen, cuja concepção é baseada no LVM do sistema HP-UX, e do outro temos o Enterprise Volume Management System (EVMS) [2] da IBM, colocado sob a GPL em 2002.

O desenvolvimento de dois diferentes gerenciadores

de volumes foi motivo de algumas discussões na comunidade Linux. Muitos questionavam se não seria melhor unir os dois projetos. Para o kernel 2.6, Linus Torvalds decidiu adotar o LVM em sua segunda versão (LVM2).

Universos paralelos

A IBM, entretanto, continua a desenvolver o EVMS. O quadro “Viagem pelo EVMS” oferece uma breve introdução desta poderosa ferramenta, que é mais do que um simples gerenciador de volumes para Linux. Este artigo, todavia, tratará mais especificamente do LVM, que normalmente já está disponível no kernel 2.4 das atuais distribuições Linux.

O kernel 2.4 padrão utiliza ainda a versão 1.x do LVM. Para atualizar o sistema

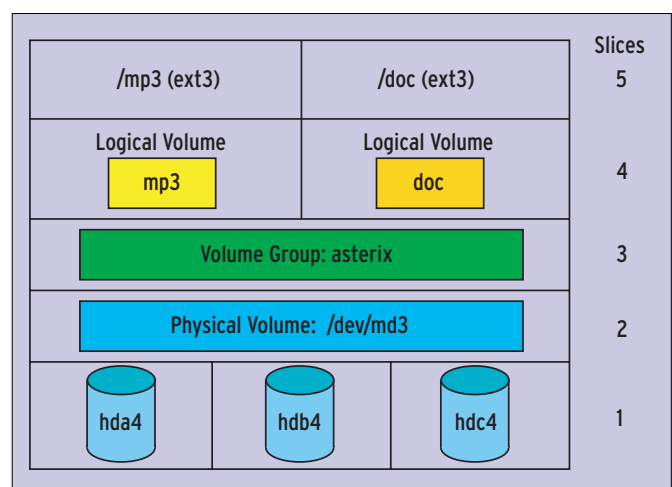


Figura 1: O grupo de volumes concatena um ou mais volumes físicos em uma unidade de dados lógica. Os volumes lógicos utilizam essa unidade de dados e são simples dispositivos de bloco (block devices).

para o LVM da versão 2.00.20 temos que aplicar a ele as alterações fornecidas pelo pacote disponível em [3]. Esse pacote contém, além dos patches para o kernel 2.4, também a versão adequada dos programas de gerenciamento do LVM, bem como as alterações necessárias do mapeador de dispositivos (aka “device mapper”) do sistema.

No que tange a gerenciadores de volume, o SuSE Linux 9.1 oferece a maior variedade de escolha entre as distribuições Linux, incluindo o LVM, LVM2 e o EVMS. O Fedora Linux inclui o LVM e LVM2 e o Red Hat dispõe atualmente apenas do LVM em suas versões Professional Workstation e Enterprise.

Os exemplos que mostraremos a seguir funcionam com a versão 1.x do LVM. As novidades da versão 2 estão descritas no quadro “Novidades no LVM2”.

LVM em Raid por Software

O sistema de trabalho do LVM é dividido em três partes: o volume físico (PV), o grupo de volumes (VG) e o volume lógico (LV). A Figura 1 esclarece a terminologia e ilustra as camadas de abstração do sistema.

A base ilustrada na camada 1 representa os discos rígidos reais - no nosso caso três discos IDE. Sobre ela, na segunda camada, é constituído o chamado “Multiple Device” (*/dev/md3*) do Raid por software - podemos dizer, entretanto, que ele é um volume físico (PV), uma vez que o Raid por software emula um dispositivo físico. No caso do Raid por hardware, a controladora Raid com-

bina os discos rígidos ilustrados na camada 1 em um único dispositivo físico de armazenagem (dispositivo de bloco).

No início de cada volume físico (PV) encontra-se, no primeiro bloco físico do volume (PE), a área de descrição do grupo de volumes (VGDA). Ela contém meta-dados da configuração do LVM, é dividida em quatro setores e pode ser comparada à tabela de partição de um disco rígido comum. No diretório */etc/lvmconf* é gravada uma cópia da VGDA.

O grupo de volumes “asterix” na camada 3 combina um ou mais volumes físicos em um único - neste exemplo apenas */dev/md3*. É como se fosse criada uma nova unidade de armazenagem que se estendesse por todos os volumes físicos. O grupo de volumes é totalmente flexível, podendo ser facilmente expandido com volumes físicos adicionais.

Na quarta camada estão os volumes lógicos: no exemplo são eles *mp3* e *doc*. Cada volume lógico (LV) é um dispositivo de blocos separado, com o nome do volume lógico correspondente, no nosso exemplo */dev/asterix/mp3* e */dev/asterix/doc*, cujo tamanho pode ser modificado posteriormente. Em ambos os volumes deve ser gerado um sistema de arquivos, que será montado no sistema como de costume (camada 5).

A unidade indivisível...

Vamos agora “dissecar” as camadas dois, três e quatro. A menor unidade de armazenagem físico de um sistema LVM é o bloco físico (PE), citado anteriormente. Cada volume físico, quando o grupo de volumes é criado, é dividido em blocos físicos de mesmo tamanho, por padrão 32 MBytes. No total, um volume físico pode conter até 65535 blocos físicos, o que resulta em um tamanho máximo de 2 TBytes. Cada bloco físico recebe uma identificação numérica (ID) que começa do 0 em cada volume físico - desse modo, cada bloco físico dentro de um volume físico tem seu próprio ID.

Analogamente, cada volume lógico (LV) é dividido em blocos lógicos (LE). O tamanho dos blocos lógicos e físicos é sempre o mesmo; um bloco lógico tem assim, por padrão, 32 MBytes de tamanho. Como no caso dos blocos físicos, os blocos lógicos são numerados a partir do 0 e estão conectados logicamente ao seu bloco físico correspondente, de modo

que podemos dizer que cada bloco lógico (LE) está mapeado exatamente contra um único bloco físico (PE). A Figura 2 mostra como os volumes lógicos *mp3* e *doc* estão mapeados no volume físico */dev/md3* do grupo *asterix*.

Quando um programa acessa um determinado byte do volume lógico *doc*, primeiramente o ID do bloco lógico no qual os dados se encontram é calculado. Por meio da correlação dos blocos lógicos com os físicos, o LVM identifica o ID do bloco físico apropriado e acessa a posição correspondente em */dev/md3* - não se esqueça que *doc* está mapeado no volume físico */dev/md3*.

Fatiado

Há dois modos de distribuição para os dados nos volumes físicos: linear e em fatias (“striping”). Normalmente, o LVM utiliza o modo linear, adequado para grupos de volume que consistem apenas de um volume físico e um disco rígido ou sistema Raid. Os blocos lógicos são associados de modo linear e crescente aos blocos físicos.

Para grupos de volume que se estendam por vários volumes físicos, a opção de distribuição em fatias torna-se interessante. Blocos lógicos vizinhos são, neste caso, associados a blocos físicos de dois volumes físicos diferentes: no caso do acesso a dados que estejam armazenados em blocos lógicos vizinhos, tais dados vêm de diferentes discos rígidos, o que em determinados casos pode aumentar a velocidade de transferência. A principal desvantagem é que esse tipo de volume lógico não pode ser expandido posteriormente. Por isso, na prática, o modo linear é quase o único a ser utilizado. O comando

```
lvcreate -L300 -nmp3 asterix
```

produz um volume lógico em modo linear de 320 MByte chamado *mp3* dentro do grupo de volumes *asterix*. Se desejarmos utilizar o modo em fatias, temos que adicionar o parâmetro *-i*, bem como o número de fatias - por exemplo *-i2*, no caso de dois volumes físicos.

O volume lógico *mp3* do exemplo acima deveria ter 300 MByte. No entanto, como um bloco lógico tem 32 MByte, a partição foi expandida automaticamente para o próximo múltiplo de 32 MByte.

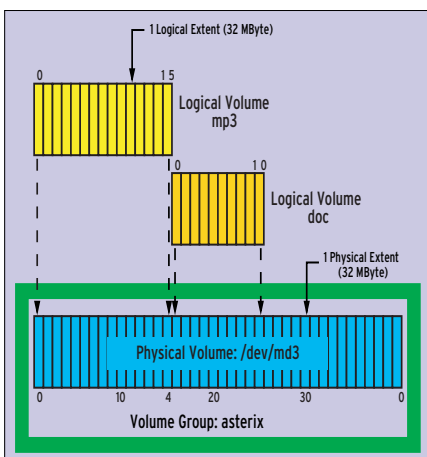


Figura 2: Os blocos lógicos têm o mesmo tamanho de seus blocos físicos correspondentes. Em modo linear, os volumes lógicos ficam diretamente um atrás do outro no volume físico.

Ferramentas

O pacote LVM contém uma gama completa de ferramentas de administração, com as quais as diferentes camadas descritas anteriormente podem ser criadas e manipuladas. Os programas *vgcreate*, *vgdisplay*, *vgchange* e *vgremove*, por exemplo, são responsáveis pela manipulação de grupos de volumes.

Para fazer um backup e restaurar os arquivos de configuração do grupo de volumes temos os utilitários *vgcfgbackup* e *vgcfgrestore*; *vgreduce* e *vgextend* diminuem ou expandem um grupo de volumes. Para transferir um grupo de volumes de um computador para outro temos *vgexport* e *vgimport*. Para dividir ou concatenar vários grupos de volumes utiliza-se *vgsplit* e *vgmerge*; *vgscan* procura por grupos de volumes perdidos em dispositivos de blocos e *vgrename* modifica seus nomes.

Para as duas outras camadas de abstração - volumes físicos e lógicos - há também ferramentas semelhantes, cujos nomes só se diferenciam dos citados anteriormente pelas duas primeiras letras: *pv* para volumes físicos e *lv* para volumes lógicos.

Planejamento

O volume físico é um array RAID nível 5 por software, composto por três discos rígidos IDE, conforme mostrado na Figura 1. Os dados de */mp3* e */doc* devem ser armazenados no volume RAID e o uso do LVM é indicado caso seja necessária uma futura expansão no espaço de armazenamento.

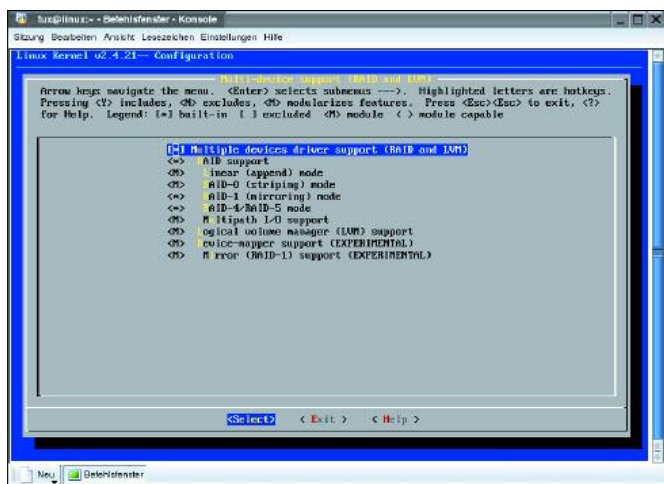


Figura 3: O kernel 2.4 ainda contém a versão 1 do Logical Volume Manager (LVM). Para utilizar o LVM2 ele precisa ser alterado com os "patches" disponíveis em [3].

Excursão pelo EVMS

O Enterprise Volume Management System (EVMS) da IBM fornece uma estrutura básica para todos as variações do gerenciamento de volumes. Sua arquitetura consiste em um modelo baseado em "plugins", no qual se podem adicionar módulos isoladamente, em forma de extensões. Ele é compatível com LVM, integra Raid por software ("Multiple Devices") e oferece suporte para os sistemas de arquivo atuais. A resetorização de blocos defeituosos ("Bad Block Relocation" - BBR) e o suporte para cluster também não são novidades para o sistema. A Figura 5 ilustra os módulos do kernel isoladamente, bem como os plugins que acompanham o kernel padrão do SuSE Linux 9.0.

Além da interface gráfica de administração

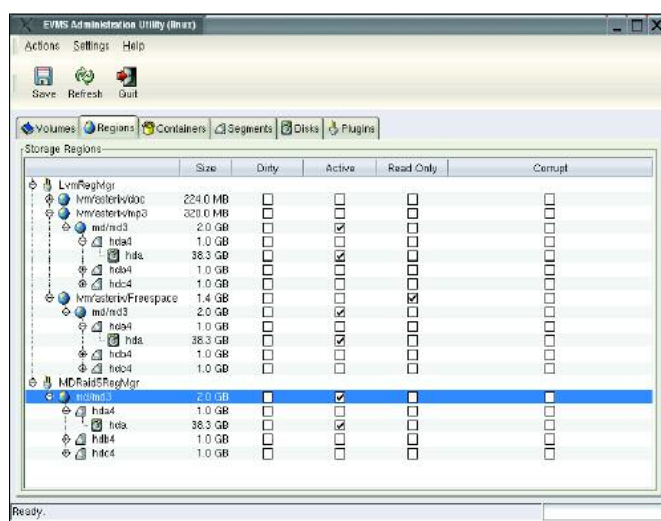


Figura 6: O EVMS pode ser administrado via interface gráfica ou em modo texto. A nomenclatura utilizada é diferente da usada pelo LVM.

do EVMS (Figura 6) há também uma interface em modo texto. A terminologia do EVMS é um pouco diferente da do LVM - volumes físicos são chamados de segmentos, grupos de volumes são conhecidos por containers e os volumes lógicos são chamados de regiões. A documentação disponível na página do projeto na Internet [2] facilita a familiarização com o sistema.

Instruções sobre a configuração de um Raid5 por software a partir de três discos rígidos são encontradas em [1]. Nos próximos exemplos partimos do pressuposto de que o sistema Raid */dev/md3* já esteja funcionando e inicializado.

No kernel do Linux na categoria *Multiple Device Support* os itens *RAID support*

e *Logical Volume Manager (LVM) support* devem estar ativados (veja Figura 3). No caso do kernel 2.4 "vanilla" precisamos ainda aplicar as alterações mencionadas no início deste artigo para atualizar o LVM para a versão 2.

Caso o suporte a LVM não esteja compilado no kernel e sim disponível na

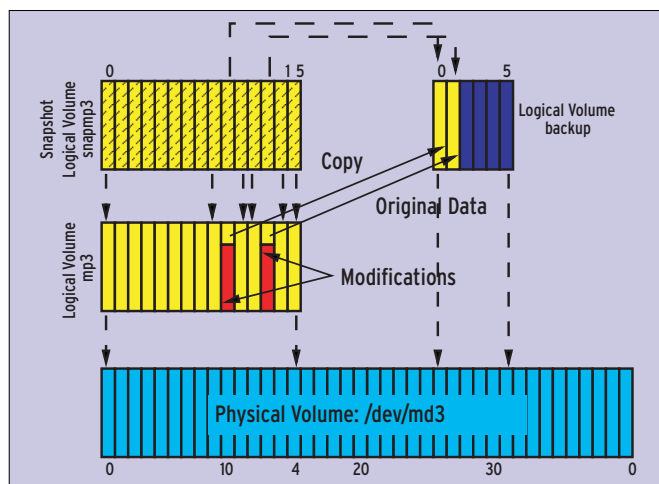


Figura 4: Um snapshot é um volume lógico apenas de leitura, que garante a consistência dos dados em um grupo de volumes para backup, embora o volume lógico original permaneça alterável.

O que há de novo no LVM2

As três principais modificações ocorridas no LVM2 são o mapeador de dispositivos (*Device Mapper*), o novo formato dos metadados e o arquivo de configuração *lvm.conf*. O novo mapeador de dispositivos permite criar um novo dispositivo de blocos em um dispositivo já existente. O formato dos metadados do LVM2 é mais estável e estruturado de maneira mais eficiente.

No novo arquivo de configuração (*lvm.conf*) é possível incluir parâmetros para cada dispositivo isoladamente, bem como ajustar o backup dos metadados e o nível de *logging* (registro) do sistema. O LVM2 é retro-compatível com o LVM1. Para converter os metadados do grupo de volumes *asterix* ao novo formato, use o comando *vgconvert -Mz asterix*.

forma de um módulo, precisamos colocá-lo na RAM Disk inicial do sistema (*initrd*). Isso pode ser feito pelo comando *lvmcreate_initrd*.

Criação dos volumes

O volume físico é o Raid5 por software */dev/md3*. O comando *pvcreate /dev/md3* prepara o dispositivo de blocos da maneira adequada. Sua configuração pode ser verificada através do comando *pvdisplay /dev/md3*.

O grupo de volumes *asterix* é criado pelo comando *vgcreate asterix /dev/md3*. No caso de vários volumes físicos, adicione o nome do dispositivo de blocos correspondente ao comando.

A quarta camada, com os volumes lógicos *mp3* e *doc*, é criada pelo comando *lvcreate -L300 -nmp3 asterix* e *lvcreate -L200 -ndoc asterix*. Com isso, o tamanho real dos volumes lógicos é de 320 MByte para o primeiro e 224 MByte para o segundo - lembre-se que o LVM utiliza somente blocos lógicos com tamanhos múltiplos de 32 MByte, como já dissemos anteriormente.

No diretório */dev/asterix* devem existir agora três novos dispositivos: o dispositivo de caracteres (“character device”) *group* e os dois dispositivos de blocos *mp3* e *doc*. Os sistemas de arquivos são criados da mesma forma que nos dispositivos de blocos comuns, com os comandos *mkfs.ext3 /dev/asterix/mp3* e *mkfs.ext3 /dev/asterix/doc*.

A montagem dos volumes lógicos no sistema de arquivos também não se dife-

rencia em nada daquela dos dispositivos de blocos comuns. Entretanto, antes de ser montados, os programas *vgscan* e *vgchange -ay* precisam ser executados. Além disso, durante o processo de desligamento do sistema, deve-se rodar o programa *vgchange -an*. De preferência, esses comandos devem ser adicionados aos scripts de inicialização e desligamento do sistema: */etc/init.d/boot* e */etc/init.d/halt*, de acordo com a distribuição Linux utilizada.

Quando o espaço do volume lógico *mp3* acabar, basta usar o comando *lvextend -L 500M /dev/asterix/mp3* para expandi-lo para 500 MByte. Em seguida, o sistema de arquivos tem que ser ajustado ao novo tamanho do volume lógico, utilizando - no caso do sistema de arquivos Ext3, por exemplo - o comando *resize2fs /dev/asterix/mp3*.

Snapshots

A função de snapshot do LVM é especialmente útil para efetuar backups de um grupo de volumes. Ela permite que se crie, a qualquer momento, um “clone” do volume lógico que não pode ser modificado diretamente. O snapshot assim criado é uma cópia “congelada” (*frozen image*) do volume lógico original, que deve ser montado apenas para leitura no sistema de arquivos durante o processo de backup. O volume lógico original permanece alterável e modificações efetuadas em qualquer de seus blocos lógicos ocasionam automaticamente a cópia do bloco lógico original para o snapshot (Figura 4).

A quantidade de memória designada para o snapshot deve ser suficiente para absorver pelo menos a mesma quantidade de dados escrita no volume lógico original durante o backup. De preferência, pode-se criar um grupo de volumes à parte, por exemplo *operator*, em um disco rígido extra. Se a quantidade de dados a serem gravados no snapshot superar o tamanho indicado durante sua criação, ou se o computador for reiniciado, o snapshot desaparece automaticamente - os dados escritos nele até então são copiados de volta para o volume lógico original.

Instruções detalhadas para a realização de backups com snapshots podem ser encontradas no tutorial de LVM [4], no capítulo 11.4.

Riscos do “fatiamento”

Dois tutoriais ([5] e [6]) descrevem como instalar um sistema de arquivos raiz em um Raid por software com LVM. Isso faz sentido em alguns casos, mas aumenta em muito o trabalho do administrador e, em caso de falha em algum dos componentes do sistema de armazenagem, as coisas podem se complicar. É importante verificar em que casos o uso do LVM faz sentido e é necessário.

Outra ponto que deve ser muito bem analisado é o emprego do modo em fatias (*striping*), pois ele nem sempre é sinônimo de melhor desempenho - além de ter a desvantagem de engessar o tamanho do volume lógico, que não pode mais ser alterado.

A falha de um disco rígido é compensada pelo sistema Raid, mas ela pode tomar proporções catastróficas se um volume lógico no modo em fatias estiver distribuído em dois ou mais discos rígidos. Neste caso o conjunto de dados inteiro torna-se inconsistente, enquanto que para volumes lógicos em modo linear apenas os dados do volume lógico afetado pela falha são atingidos por ela. ■

Legenda

LE	Bloco Lógico
LV	Volume Lógico
LVM	Logical Volume Manager
PE	Bloco Físico
PV	Volume Físico
VG	Grupo de Volumes
VGDA	Descritor do Grupo de Volumes

INFORMAÇÕES

[1] Carsten Wiese, “Volks-Raid”: Linux-Magazin 07/2003, página 50; <http://www.linux-magazin.de/Artikel/ausgabe/2003/07/volksraid/volksraid.html>

[2] Enterprise Volume Management System (EVMS): <http://evms.sf.net>

[3] Patch e mapeador de dispositivos para o LVM2: <http://sources.redhat.com/pub/lvm2/>

[4] Tutorial de LVM: <http://ldp.org/HOWTO/LVM-HOWTO/>

[5] Tutorial de implementação de um sistema raiz com LVM e Raid por software: <http://www.midhgard.it/docs/lvm/html/index.html>

[6] Tutorial de implementação de um sistema raiz em Raid por software: <http://karaolides.com/computing/HOWTO/lvmraid>

Patrick Volkerding ataca novamente

A Bola da Vez

Vamos começar pelos números: GNOME 2.6, KDE 3.2.3, Gimp 2.0.2, kernel 2.4, GCC 3.3.4 e vários outros pacotes compõem o Slackware 10, que já está preparado para o kernel 2.6, com pacotes pré-compilados da versão 2.6.7 disponíveis aos usuários mais corajosos. **POR PITER PUNK**



Antes de escrever esta análise, instalei o Slackware 10.0 em oito computadores diferentes. Três deles sofreram um “upgrade” da versão 9.1 para a 10 com o utilitário *slackpkg*. As restantes passaram por uma instalação “limpa”, tanto via rede como através de CDs. Nenhuma dessas cinco máquinas teve problemas durante a instalação, que continua seguindo o mesmo padrão da versão 9.1. Em duas delas, a partição com os dados dos usuários, */home*, foi preservada.

Para instalação via rede foi necessário criar quatro discos de boot: um com o kernel (*bare.i*), dois discos de root (*install.1* e *install.2*) e um disco com módulos de rede (*network.dsk*). Uma destas máquinas utiliza discos SCSI, o que exigiu a criação de um outro disco com um

kernel adequado (*adapt.ec.s*). Se sua máquina não consegue inicializar pelo CD do Slackware 10, crie um disco de boot a partir da imagem *sbootmgr.dsk*, encontrada no diretório *isolinux* do CD. Isso foi necessário em uma das máquinas de teste.

O instalador detectou automaticamente as placas de rede PCI em todas as máquinas que as possuíam. Na máquina com uma placa de rede ISA foi necessário incluir o módulo manualmente, com informações de *io* e *irq*. Não foi preciso nenhuma configuração especial para a máquina com teclado USB, que foi reconhecido normalmente pelo instalador. Ainda no item teclado, o instalador oferece a oportunidade de configuração com os layouts ABNT2 e International, os mais usados no Brasil.

A instalação completa ocupa entre 2.5 e 3 Gigabytes de espaço em disco, mas é possível reduzir seu tamanho sem muito esforço, escolhendo apenas as séries e pacotes necessários às suas necessidades. Se não quiser ter problemas com bibliotecas, lembre-se de instalar todos os pacotes da série *L*.

Recém-nascido

O primeiro boot corre normalmente; todos os serviços selecionados durante a instalação são iniciados corretamente. Na verdade, o MySQL retornou um erro, mas o arquivo */etc/rc.d/rc.mysql.d* diz como ele deve ser solucionado. Em algumas máquinas pode surgir um *WARNING* (alerta) de que os canais de som estão mudos. Nesse caso, é necessário ajustar manualmente o volume destes canais com o utilitário *alsamixer*.

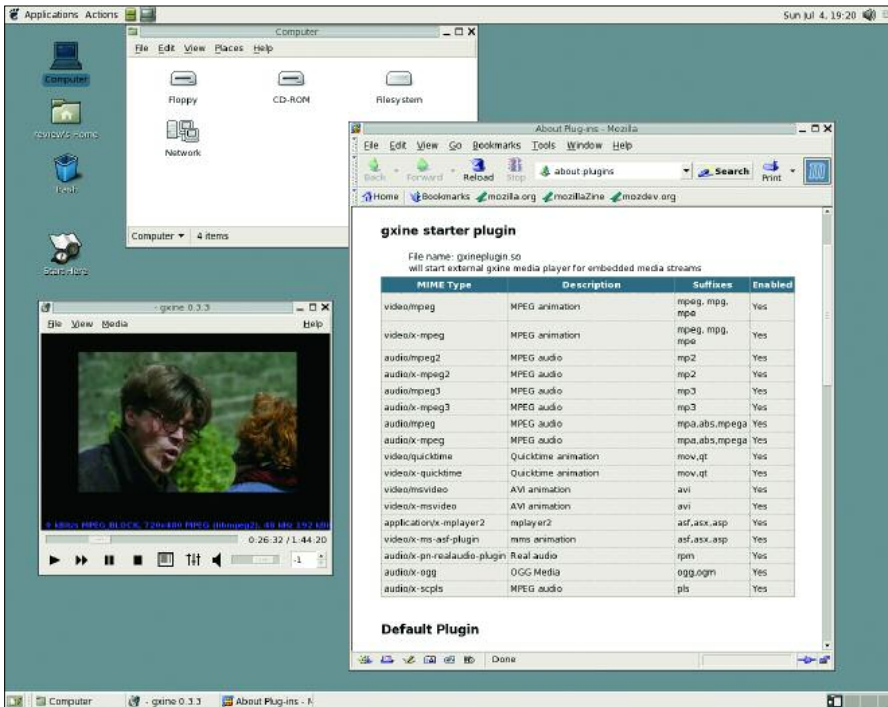
No primeiro login (que deve ser feito com o usuário *root*) o usuário é informado de que tem novas mensagens na caixa postal. São duas, para ser exato: uma recomenda o cadastro da máquina no *Linux Counter* [3] e a outra, do criador do Slackware, Patrick Volkerding, agradece a preferência e dá diversas dicas sobre a configuração de gravadores de CD, impressoras, compartilhamento de arquivos com NFS etc.

No papel e na rede

Dois sistemas de impressão estão disponíveis, o LPRng (veja artigo na página 50) e o CUPS (veja nossa primeira edição). Para configurar o LPRng, a melhor opção é utilizar o *apsfilter*. Já o CUPS possui o seu próprio utilitário de configuração via web. Numa instalação completa, o gerenciador de impressão padrão é o LPRng.

Em todas as máquinas, a configuração de rede feita durante a instalação foi preservada. Todas as placas de som PCI foram corretamente reconhecidas, mas foram justamente estas as máquinas que apresentaram um “WARNING” sobre o volume. Resolver isso é fácil: execute o *alsamixer*, tire os canais de som do mudo, aumente o volume e digite *alsactl store* para guardar as configurações.

O próximo passo é iniciar o ambiente gráfico. O Slackware 10, como muitas outras distros recentes, não usa mais o tradicional XFree86, e sim seu “fork”, o X.org (ou simplesmente “X”). Com isso,



Nautilus, com o ícone "Computer" selecionado, mostrando os dispositivos acessíveis ao usuário e o ícone de rede. O gxine está sendo usado para exibir um DVD.

vários arquivos e utilitários que já estávamos habituados a usar mudaram de nome. As principais diferenças são mostradas na Tabela 1.

Pixels de montão

O Slackware já vem com uma configuração "padrão" do X, utilizando o driver de vídeo *vesa*, teclado *us* e mouse PS/2. Se você utilizar algo diferente disso, deve editar o arquivo *xorg.conf* ou usar um dos utilitários de configuração mencionados anteriormente.

O servidor X foi configurado com o utilitário *xorgconfig*, com ajuste fino através da edição do arquivo *xorg.conf*. Nas máquinas com aceleradoras 3D da nVidia a instalação do driver, disponível no site do fabricante, foi simples: basta executar o binário e seguir as instruções na tela. As outras máquinas utilizaram os drivers do próprio projeto X.org.

E justamente em nossos testes com o X tivemos o primeiro problema: o driver *vesa* congelou duas das máquinas de

teste. Como as máquinas estavam em rede, não foi difícil nos conectarmos ao sistema remotamente e recuperar o controle. As placas de vídeo com problema foram uma nVidia GeForce e uma SiS6326. Todas as outras máquinas, inclusive uma com outra SiS6326, funcionaram normalmente.

Ratos e outros bichos

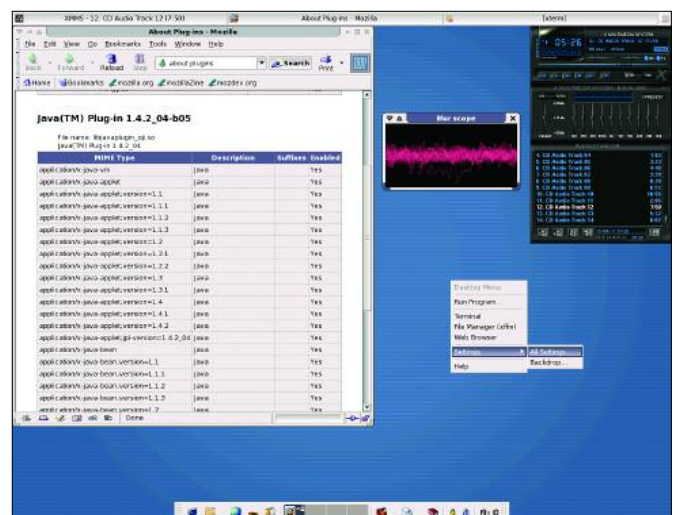
Utilizamos diversos mouses, tanto modelos PS/2 quanto USB, quase todos óticos e com a famosa "rodinha" de rolagem. Uma das máquinas utiliza um mouse PS/2 antigo da Compaq e outra um mouse serial. É importante lembrar que o utilitário *mouseconfig* apenas configura o mouse no modo texto e que o arquivo de configuração do servidor X padrão vem preparado apenas para mouses PS/2. Você deverá modificar isso caso queira usar um mouse serial ou USB no modo gráfico. Entre os

teclados que configurei para uso no ambiente gráfico havia modelos DIN, PS/2 e USB, com "layouts" US Internacional, ABNT2 e até mesmo Francês.

Ambiente de trabalho

Existem gerenciadores de janelas para todos os gostos: *mwm* (0.93.94), *fwm2* (2.4.18), *fwm95* (2.0.43b), *WindowMaker* (0.80.2), *fluxbox* (0.1.14) e *blackbox* (0.65.0), além dos ambientes de trabalho *GNOME* (2.6), *KDE* (3.2.3) e *XFCE* (4.0.5). A política da distribuição é não mexer nos menus e outros detalhes cosméticos, deixando o usuário com uma configuração praticamente idêntica à que teria se houvesse instalado o software manualmente. Infelizmente, isso faz com que os gerenciadores de janelas e ambientes gráficos fiquem com configurações bem irregulares, o que pode deixar o usuário com uma má impressão sobre o ambiente escolhido.

Os menus do WindowMaker são compostos praticamente por aplicativos inexistentes (ou não usados), enquanto os do KDE são extremamente poluídos e confusos. Já os do fluxbox e blackbox são bastante úteis. Os do xfce também são bons, mas extremamente "curtos" (o que parece fazer parte da filosofia do projeto). Se o usuário do WindowMaker ou do XFCE não personalizar o seu menu (adicionando itens, ícones ao dock, etc...), será necessário sempre manter um terminal aberto. Entretanto, a configuração padrão do GNOME está muito bem-feita, mostrando a preocupação dos envolvidos no projeto em oferecer uma interface limpa e funcional.



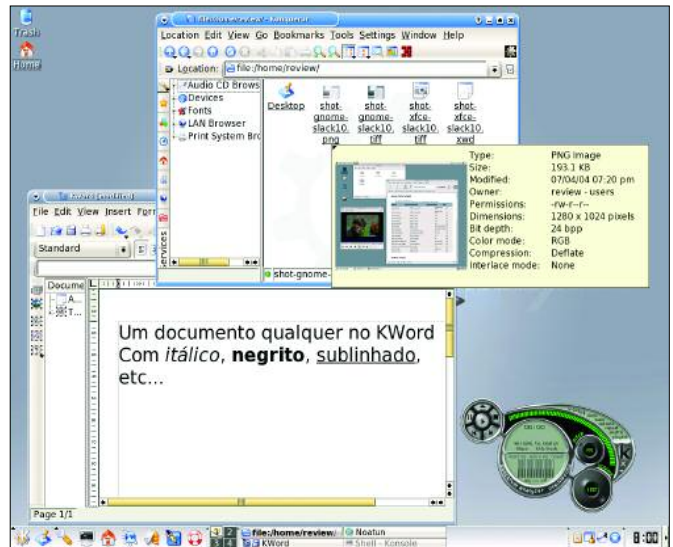
O gerenciador de janelas XFCE, com o menu padrão.

Tabela 1

Antigo nome	Novo nome
<i>/etc/X11/XF86Config</i>	<i>/etc/X11/xorg.conf</i>
<i>xf86confi</i>	<i>xorgconfig</i>
<i>xf86cfg</i>	<i>xorgcfg</i>
<i>xfree86setup</i>	<i>xorgsetup</i>



Gimp, versão 2.0.2, rodando no Window Maker.



Alguns dos aplicativos do KDE 3.2: Konqueror, Kword e o media player noatun.

Navegando

Para quem faz a instalação completa, o Mozilla (versão 1.7, a nova versão de referência para os produtos da Mozilla Foundation) já vem configurado com os plugins necessários para ver páginas em Flash e rodar Applets Java. Quem não gosta do Mozilla pode utilizar outros browsers, como o *epiphany*, *galeon* ou *konqueror*. Os que preferem navegar em modo texto têm à sua disposição o *lynx* e o *links*, sendo que o último foi compilado com suporte à exibição de imagens, usando para isso a *svglib* ou o *framebuffer* do console.

Multimídia

Para tocar arquivos de áudio temos, além de vários players em modo texto como o *workbone*, *ogg123*, *madplay*, *mpg123* e outros, o *noatun* (com várias skins e plu-

gins de visualização) e o tradicional *xmms*. Usando o *xine* é possível assistir a DVDs e outros formatos de vídeo (e também reproduzir arquivos de áudio). Para tocar DVDs “comerciais” (com trava de região), é necessário baixar e instalar a *libdvdcss*, responsável por “decodificar” os discos. O *gxine*, que utiliza o *xine* para reproduzir arquivos de vídeo e áudio, também serve para assistir vídeos na Web. Ele se auto-configura como um plugin para o Mozilla na primeira vez em que é executado.

A gravação de CDs também ficou mais simples: ela pode ser feita diretamente a partir do Nautilus, gerenciador de arquivos do Gnome, graças ao pacote *nautilus-cd-burner*. Também há os utilitários *k3b* e *xcdroast* no diretório */extra*, para aqueles discos mais complexos. Na linha de comando estão disponíveis os indispensáveis *cdrecord* e *cdrdao*.

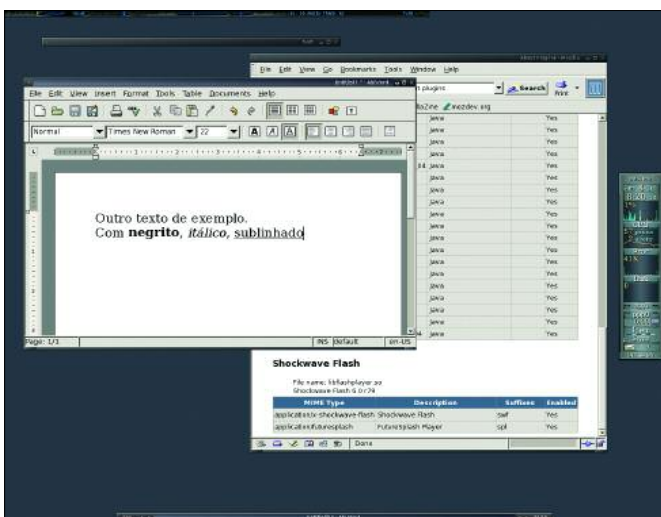
Para o administrador ele é uma importante ferramenta para monitoramento de servidores em tempo real, composta pelo daemon *gkrellmd* e alguns clientes espalhados pela tela. Assim, é possível monitorar de maneira fácil várias máquinas, além de configurar alarmes para estados críticos em qualquer uma delas.

Um motivo de alegria para os administradores de rede é a entrada oficial do *iproute2* na distribuição. Com isso, uma série de novos recursos de roteamento, criação de túneis, qualidade de serviço (QoS) e balanceamento de carga torna-se disponível. Outra importante adição é o *dnsmasq*, um servidor DNS especial para redes mascaradas. Além de servidor DNS, o *dnsmasq* funciona também como um servidor DHCP, diminuindo a quantidade de configurações a serem feitas ao adicionar uma nova máquina à rede.

O Slackware 10 ainda mantém no CD o diretório */pasture*, onde há alguns pacotes que não fazem mais parte da distribuição, mas que ainda tem utilidade, como o XFree86 3.3.6 (para placas de vídeo não mais suportadas pelo x.org) ou versões antigas do Berkeley DB (a distro inclui a versão 4.2.52).

De olho

Uma adição interessante, tanto para os usuários domésticos como para os administradores de sistemas, é o versátil *gkrellm*. Para os usuários ele é uma forma de monitorar a rede, montar e desmontar dispositivos ou como um simples acessório estético.



Gerenciador de janelas FluxBox. O gkrellm exibe várias estatísticas sobre a máquina.

INFORMAÇÕES

- [1] <http://www.slackware.com>
- [2] <http://www.x.org>
- [3] <http://www.counter.li.org>
- [4] <http://www.alsa-project.org>
- [5] <http://www.gnome.org>
- [6] <http://www.xfce.org>

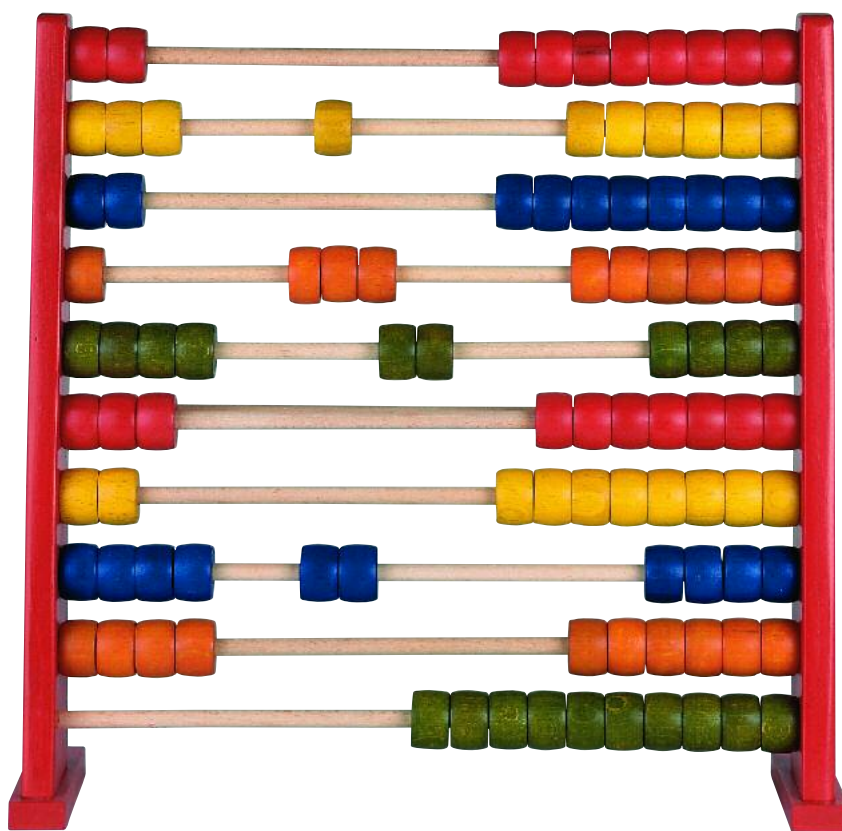
PlanMaker

A Arte das Planilhas

Depois de adaptar seu processador de texto TextMaker para Linux no ano passado, a SoftMaker agora lança seu aplicativo para planilhas, o PlanMaker.

Demos uma olhada na versão de pré-lançamento, que está quase concluída.

POR TIM SCHÜRMAN



A SoftMaker [1] é uma fabricante de softwares com diversos produtos para escritório em seu portfólio. Além do processador de texto TextMaker e do banco de dados DataMaker (que no momento está disponível apenas em versão Windows), o portfólio de produtos inclui o PlanMaker, uma planilha eletrônica que recentemente ganhou uma versão para Linux.

Assim como o editor de texto, o PlanMaker é um aplicativo extremamente pequeno, que ocupa meros 15 MBytes de espaço no disco rígido – incluindo o manual formatado em PDF, que também serve como arquivo de ajuda. O fabri-

cante não oferece ajuda on-line, em contraste com a maioria dos outros programas. Na versão trial que testamos faltavam os modelos e documentos de amostra. Em compensação, havia um grande número de dicionários de francês, português e algumas línguas escandinavas. Se todos forem instalados, o espaço utilizado no disco rígido salta para cerca de 31 MBytes.

A pouca exigência de espaço no disco rígido é, felizmente, característica do aplicativo. Se você está familiarizado com pacotes de escritório na escala do OpenOffice, ficará admirado com a velocidade com que o aplicativo é iniciado e

reage aos comandos. Mesmo assim, houve demoras ocasionais, porém perceptíveis, ao importar documentos.

Quanto à estabilidade da aplicação, não podemos nos queixar. O programa não caiu nenhuma vez em nossos laboratórios, apesar de ter havido um ou dois erros de exibição. De novo, especificamente ao importar documentos.

GUI

A interface gráfica é bastante familiar (ver Figura 1). A SoftMaker utiliza sua própria **biblioteca gráfica** (toolkit), que apareceu pela primeira vez quando o TextMaker foi lançado. As barras de ferramentas e estruturas de menu seguem os padrões atuais, o que facilita a migração para o PlanMaker.

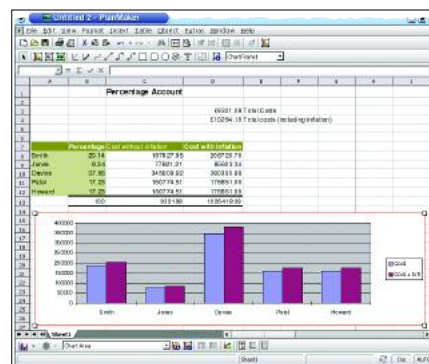


Figura 1: A interface do PlanMaker. Vários tipos de gráficos estão disponíveis.

O chamado “modo objeto” do PlanMaker demanda alguma familiaridade. O programa entra em “modo objeto” automaticamente sempre que o usuário insere ou modifica um gráfico ou imagem. Isso evita mudanças inadvertidas no conteúdo das células, mas pode irritar qualquer um que esteja acostumado a trabalhar com outras planilhas. Os usuários do TextMaker, entretanto, já devem estar habituados a essa peculiaridade. Felizmente, a maioria dos usuários não enfrentará o “modo objeto” com tanta frequência, uma vez que o próprio aplicativo se encarrega de habilitar e desabilitar automaticamente este modo sempre que necessário.

Funcionalidade

O PlanMaker recebe créditos por oferecer uma seleção generosa de elementos gráficos para construir ilustrações. É possível até mesmo agrupar fluxogramas quase instantaneamente clicando nas linhas de conexão. Elas se ligam aos objetos e se movem automaticamente com eles. Infelizmente, é preciso abrir uma caixa de diálogo (dentro de *Object* ⇨ *New Drawing* ⇨ *AutoShapes*) para acessar esse tipo de elementos especiais (ver Figura 2), mas a maioria dos usuários ficará bastante satisfeito com essa organização, já que ilustrações desse tipo são tarefas pouco comuns.

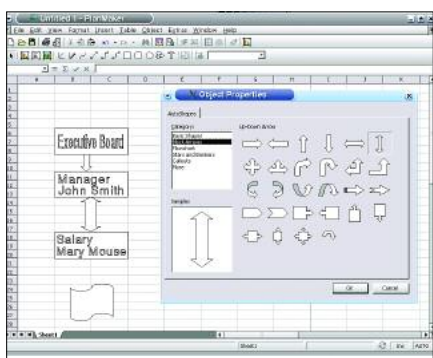


Figura 2: Ao contrário dos símbolos mais utilizados, é preciso acessar uma caixa de diálogo para selecionar elementos do fluxograma.

Fora isso, o PlanMaker é uma solução simples, mas sólida como uma rocha, embora não possa acompanhar o ritmo estabelecido pelo Open ou o StarOffice. As únicas grandes vantagens que o PlanMaker tem a oferecer são o **Goal Seeker**, uma função que encontra os valores originais para um dado resultado por aproximação, e a capacidade de agrupar células arbitrárias. No programa faltam muitas das convenientes funções que o OpenOffice tem para aliviar o esforço de trabalhar com planilhas maiores. Um exemplo é o “detective” do OpenOffice, que pode marcar todas as linhas modificadas pela fórmula contida em uma dada célula. Qualquer aplicativo moderno de planilha deve ter funções extremamente úteis como essa.

Detalhes

O problema está nos detalhes, como na maioria dos casos. O corretor ortográfico não apenas sublinha as palavras mal soletradas, mas interrompe o usuário continuamente abrindo caixas de mensagem.

Os usuários de Excel podem, também, sentir falta de ajuda ao digitar os dados, como o “autocompletar” automático.

A seleção de formatos pré-definidos de células do PlanMaker mostra o programa sob uma luz mais favorável. Há até mesmo modelos para códigos ISBN. A planilha pode também transpor matrizes, ou seja, trocar as linhas e colunas. Enquanto o OpenOffice.org exige que os usuários façam um passeio pelo clipboard, o PlanMaker tem um item no menu principal que realiza a transposição diretamente.

Importando

Os sentimentos voltaram a ser antagônicos ao tentarmos importar documentos do Excel. Embora o PlanMaker tenha convertido os dados de maneira bastante satisfatória, muitos dos gráficos embutidos simplesmente desapareceram (ver Figura 4). Como o programa não suporta macros ou scripts, ambos são descartados sem sequer um “até logo”. Ficamos muito contentes em ver que ele suporta arquivos do StarCalc - mas a alegria não durou muito. Na verdade, o PlanMaker só trabalha com arquivos das obsoletas versões 1.0 e 2.0. Esse problema é agra-

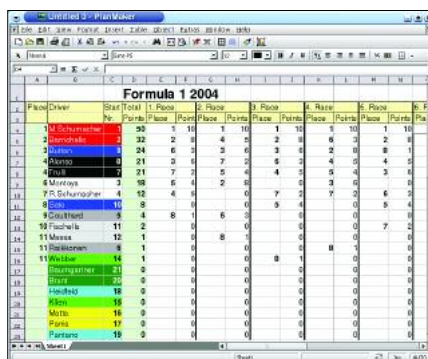


Figure 4: Um arquivo importado do Excel. O que você não vê é que o programa jogou fora a figura de um carro de Fórmula 1 que estava no fundo.

vado pelo fato de que as versões atuais do Open/StarOffice (1.2 e 7, respectivamente) não têm mais filtros de exportação para esses formatos.

Conclusão

Trabalhar com o PlanMaker é divertido. O programa é pequeno, econômico e, de fato, rápido. E mais, está disponível para várias plataformas e sistemas operacionais, incluindo dispositivos móveis como o PocketPC, além da plataforma Windows comum. Nossas reclamações em relação a ele são a falta de funcionalidade, ponto importante se você pretende usar o programa para trabalhos rotineiros de escritório, e o fato de que os usuários poderiam ter mais recursos ao trabalhar com planilhas maiores. O PlanMaker é recomendado para uso doméstico ou para máquinas mais antigas, com menos força na CPU.

Resta falar do preço. O programa, é claro, vale os US\$ 49.90 que custa. Você pode conseguir um desconto se optar pelo pacote que inclui o TextMaker, por apenas US\$ 69.90. Todavia, o OpenOffice oferece mais e é de graça, desde que você tenha o poder de processamento extra que o pacote demanda.

Se você quiser testar o PlanMaker, o Suse Linux 9.1 Professional Package contém uma edição grátis de uso ilimitado, baseada em uma versão beta (protótipo) do software. Vale também conferir a página da SoftMaker em [2] para fazer o download de uma versão de demonstração (*trial*) válida por 30 dias. ■

GLOSSÁRIO

Biblioteca gráfica: Uma coleção de ferramentas e bibliotecas de programas utilizadas para programar aplicativos ou simplificar esse processo. No caso de interfaces, as bibliotecas gráficas contêm normalmente modelos para janelas, botões ou listas comuns, influenciando bastante a aparência e a forma de um aplicativo.

Goal seeker: O Goal seeking é um método de aproximação. Se uma célula regida por uma fórmula depende de outras células, é possível especificar um valor-alvo (goal) e o PlanMaker utilizará técnicas de aproximação para descobrir valores apropriados para uma das outras células, apresentando o resultado no campo designado.

INFORMAÇÕES

- [1] SoftMaker:
http://www.softmaker.de/index_en.htm
- [2] Versão teste do PlanMaker:
http://www.softmaker.de/pmtrialreg_en.htm

Crie pacotes oficiais de software para o Slackware

Embrulha pra presente!

No último artigo, aprendemos a gerenciar pacotes no Slackware e até nos aventuramos a criar alguns com o auxílio do utilitário checkinstall. Neste artigo, vamos estudar como os pacotes são montados na distribuição principal, desbravando os SlackBuilds. **POR SULAMITA GARCIA**



mato *nome-do-software.slackbuild*.

Para demonstração, vamos utilizar um pacote simples, como o *irssi*. Ele está disponível no servidor oficial (<ftp://ftp.slackware.com/slackware-10.0/source/n/irssi/irssi.SlackBuild>). Além de simples, ele é um pacote que foi incluído há pouco tempo na distribuição, com definições mais recentes.

Como é um shell script, o SlackBuild do *irssi* começa com a definição do shell e de algumas variáveis:

```
#!/bin/sh
CWD=`pwd`
TMP=${TMP:-/tmp}
PKG=${TMP}/package-irssi
```

Estas linhas indicam o *sh* como shell, definem a variável *CWD* (*Current Work Directory* - Diretório Atual de Trabalho) como o diretório atual, a variável *TMP*, se já não estiver definida, aponta para */tmp*, e *PKG* recebe o valor *TMP/package-nome-do-software*, em nosso caso, *irssi*. Logo a seguir vêm as definições das variáveis de compilação. Veja a Listagem 1.

Desde o Slackware 9, se a variável *ARCH* (arquitetura) não estiver definida, os pacotes são compilados para a arquitetura 486. Se estiver, o script vai definir algumas flags de compilação e guardá-las dentro da variável *SLKCFLAGS*.

Posso otimizar o

código para minha máquina, que tem um processador Athlon XP, trocando *mcpu=i386* por *mcpu=athlon-xp*:

```
if [ ! -d $TMP ]; then
  mkdir -p $TMP
fi
rm -rf $PKG
mkdir -p $PKG
cd $TMP
rm -rf irssi-$VERSION
tar xjvf $CWD/irssi-$VERSION.tar.bz2
```

Aqui o script verifica se o diretório temporário definido na variável *TMP* existe e remove eventuais resíduos de uma compilação anterior. Em seguida, o código-fonte é descompactado neste diretório.

```
cd irssi-$VERSION
chown -R root.root .
find . -perm 777 -exec chmod 755 {} \;
find . -perm 664 -exec chmod 644 {} \;
```

Após entrar no diretório do código-fonte, o script realiza uma verificação de permissões, retirando as permissões de

Listagem 1

```
VERSION=0.8.9
ARCH=${ARCH:-i486}
BUILD=3

if [ "$ARCH" = "i386" ]; then
  SLKCFLAGS="-02 -march=i386 -mcpu=i686"
elif [ "$ARCH" = "i486" ]; then
  SLKCFLAGS="-02 -march=i486 -mcpu=i686"
elif [ "$ARCH" = "s390" ]; then
  SLKCFLAGS="-02"
elif [ "$ARCH" = "x86_64" ]; then
  SLKCFLAGS="-02"
fi
```

SlackBuilds são scripts que automatizam a criação de pacotes no formato oficial do Slackware. São *shell scripts* (para saber mais sobre shell scripts, veja a série "Papo de Botequim", na página 87 desta edição) que, apesar de padronizados, variam ligeiramente de acordo com as necessidades do software a ser instalado. Criar um SlackBuild facilita muito a tarefa de manutenção de futuras versões do pacote.

Os SlackBuilds estão disponíveis em qualquer servidor ftp que contenha a árvore de pacotes do Slackware. Nessa árvore há o diretório *source*, que contém o código-fonte de tudo que compõe o Slackware. Dentro do diretório de cada pacote há seu código-fonte, um arquivo com a descrição e um SlackBuild, no for-

acesso dos usuários ao código-fonte e dando acesso apenas para o *root*. A seguir vem a configuração do pacote propriamente dita:

```
CFLAGS="$SLKCFLAGS" \
./configure \
--prefix=/usr \
--sysconfdir=/etc \
--enable-ipv6 \
--with-textui \
--with-bot \
--with-proxy
make
make install DESTDIR=$PKG
```

As opções de compilação são passadas ao gcc através da variável *CFLAGS*. Aqui você pode adicionar ou remover opções: se você não quer suporte a *ipv6*, remova a linha correspondente. Se quiser testar a *garbage collection*, deve adicionar uma linha especificando isso (*--with-gc* \). Para uma lista com todas as opções do pacote, digite *./configure -help*.

Depois de tudo isso, o pacote é compilado e os binários, páginas de manual, documentação e demais arquivos serão instalados no diretório especificado na variável *DESTDIR*, como se estivessem na raiz do sistema. Se você observar o que foi instalado, notará que os diretórios */usr*, */etc* e outros foram criados em *DESTDIR*.

Na Listagem 2, o dono e o grupo dos binários são modificados para usuário *root* e grupo *bin*. Em seguida o script procura os binários no formato ELF e realiza um *strip*. Isto vai "limpar" o executável, removendo símbolos de depu-

Erramos!

No artigo "Embrulha pra Viagem!", na página 55 de nosso primeiro número, um erro de edição prejudicou a compreensão de um trecho do texto. Após correção, o primeiro parágrafo da página 55, fica:

O arquivo do install.sh também não é específico ao Slackware. Ele é genérico, então para criar um pacote no formato oficial devemos substituí-lo pelo doinst.sh. Não realizar estas alterações não implica no mau funcionamento do pacote, só significa que ele não está de acordo com o padrão. É neste momento que você pode incluir tarefas no shell script doinst.sh. Nele você pode executar comandos no sistema, adicionar usuários, criar links, etc.

Listagem 2

```
chown -R root.bin $PKG/usr/bin
( cd $PKG
  find . | xargs file | grep "executable" | grep ELF | cut -f1 -d : | \
  xargs strip --strip-unneeded 2> /dev/null
  find . | xargs file | grep "shared object" | grep ELF | cut -f1 -d : | \
  xargs strip --strip-unneeded 2> /dev/null
)
mv $PKG/etc/irssi.conf.$PKG /etc/irssi.conf.new
```

Listagem 3

```
rm -rf $PKG/usr/share/doc
( cd $PKG
  find . -name perllocal.pod | xargs rm -f
)
mkdir -p $PKG/usr/doc/irssi-$VERSION
cp -a AUTHORS COPYING INSTALL NEWS README TODO docs $PKG/usr/doc/irssi-$VERSION
rm -f $PKG/usr/doc/irssi-$VERSION /docs/Make* \
$PKG/usr/doc/irssi-$VERSION/docs/*.1
```

ração, o que diminui o tamanho dos arquivos. Logo após, o script muda a extensão do arquivo de configuração para *.new*, para não sobrescrever algum arquivo de configuração já existente durante a instalação. Continuando nossa análise, veja o código na Listagem 3.

Lá o script faz alguns ajustes na documentação, movendo o que realmente interessa para um diretório com um nome significativo, composto pelo nome e versão do pacote.

```
gzip -9 $PKG/usr/man/man?/*.*
```

A linha acima comprime as páginas de manual (padrão no Slackware, lembra?).

```
mkdir -p $PKG/install
cat $CWD/slack-desc > $PKG/install\
/slack-desc
zcat $CWD/doinst.sh.gz > $PKG\
/install/doinst.sh
```

Aqui são criados os arquivos *slack-desc* e *doinst.sh*, discutidos no primeiro artigo. Recapitulando: o *slack-desc* é um arquivo de descrição do pacote e *doinst.sh* é um shell script que irá realizar tarefas pós-instalação, como a criação de links, novos usuários e ajustes necessários ao bom funcionamento do pacote.

A localização desses arquivos pode ser

modificada e o *zcat* pode ser substituído pelo *cat*, caso o arquivo *doinst.sh* seja um shell script sem compactação.

```
cd $PKG
makepkg -l y -c n $TMP/irssi-\
$VERSION-$ARCH-$BUILD.tgz
```

Finalmente podemos criar o pacote com o *makepkg*. As opções significam:

- *-l y*: se um link simbólico é encontrado, uma instrução para criá-lo é adicionada ao script *doinst.sh*. Em seguida, o link é removido.
- *-c n*: se não for *n* (no), o *makepkg* irá configurar todas as permissões de acesso ao diretório para 755 e o dono para *root.root*. Não é o que queremos.

Agora você já sabe como funciona um script que cria um pacote para o Slackware. Que tal utilizar esse conhecimento para "empacotar" um software que ainda não faz parte da distribuição? ■

SOBRE A AUTORA

Sulamita Garcia é formada em Ciências da Computação pela UFSC, onde conheceu o Linux. Participa do projeto LinuxChix e mantém o site de Alta Disponibilidade na UnderLinux, possuindo também certificação LPIC II. Trabalha atualmente na Cyclades como Software Designer.



Configurando um sistema de impressão

LPRng - A nova geração

O servidor de impressão mais popular no mundo *NIX é o lpd, utilizado desde tempos longínquos. Seu sucessor, o LPRng, segue a mesma filosofia (e forma de configuração), mostrando-se um sistema confiável, leve e rápido para gerenciar e utilizar impressoras. **POR PITER PUNK**



Yuri Maslov - www.sxc.hu

A filosofia UNIX dita o uso de vários pequenos programas, cada qual cuidando muito bem de uma função específica, trabalhando em conjunto para realizar grandes feitos. Um bom exemplo disso é o sistema de impressão no Linux.

No nível mais baixo, temos o kernel, que trabalha diretamente na comunicação com a impressora. É nesse nível que vamos encontrar o suporte à porta paralela, serial, USB e à rede. Obviamente, se você pretende utilizar uma impressora que se conecta à porta paralela, irá precisar do suporte a esse tipo de dispositivo no kernel. A maioria das distribuições Linux já fornece este recurso, se não compilado no kernel, como um módulo carregado automaticamente sempre que necessário.

Logo acima do kernel, encontramos o filtro de impressão. Este programa irá traduzir o documento a ser impresso para a linguagem específica da impressora. Sim, as impressoras possuem uma linguagem própria e os documentos devem ser traduzidos para esse idioma que, via de regra, muda de fabricante para fabricante e, dentro de um mesmo fabricante, de modelo para modelo. O filtro cuida disso, convertendo os documentos recebidos para algo que a impressora entenda.

Quem recebe o documento e o envia para o filtro de impressão é o daemon de impressão. Pode ser o BSD LPD, o CUPS ou o LPRng, entre outros. A função destes daemons é receber o trabalho enviado pelo usuário e enviá-lo para a impressora correta através do filtro cor-

reto. Se for uma impressão local, ele deve receber o documento, passá-lo pelo filtro e enviar o resultado para o dispositivo de impressão. Caso seja uma impressora remota, ele recebe o documento e conversa com o daemon da máquina remota, enviando-o para ela. Algumas vezes, o filtro é executado na máquina de origem do trabalho e, em outras, na máquina de destino.

O último elo dessa cadeia são os comandos do usuário, que enviam os trabalhos para o daemon e recolhem dele informações sobre o andamento da fila de impressão e o que está acontecendo com cada trabalho.

Neste artigo iremos configurar uma impressora utilizando o LPRng. Veremos quais os arquivos envolvidos, os parâmetros mais importantes e criare-

mos um pequeno filtro de impressão. Existem ferramentas que facilitam e automatizam boa parte desse processo, mas nenhuma delas oferece todos os recursos e tampouco a flexibilidade que o conhecimento do sistema como um todo pode trazer.

No princípio...

Era o vazio, e os usuários mandavam seus dados diretamente para os dispositivos. O *cat* reinava soberano. Para enviar um documento para a impressora bastava ordenar:

```
# cat documento > /dev/lp
```

E o documento era impresso. As impressoras daquela época lidavam apenas com texto, mas mesmo assim podiam estar em outra máquina, e por isso existiam diferentes formas de imprimir. Para resolver esses problemas surgiu o *lpd*, que lê no arquivo */etc/printcap* onde está a impressora e qual filtro de impressão deve ser usado.

Nossa primeira impressora

É extremamente simples configurar uma impressora no BSD LPD ou no LPRng. Basta colocar os dados referentes à ela no arquivo */etc/printcap*. A seguir, um exemplo de como é a entrada de uma impressora neste arquivo:

```
lp|printer1:\
:lp=/dev/lp0:\
:sd=/var/spool/lpd/lp:\
:mx#0:\
:sh:
```

A primeira coisa que temos é o nome da impressora (*lp*), seguido do apelido dado a ela (*printer1*), separado pelo caractere *pipe* (*|*). A partir daí, temos o restante da configuração, com cada um dos campos separado por “:”

- *lp=/dev/lp0*: aqui definimos qual o dispositivo da impressora. */dev/lp0* é a primeira porta paralela.
- *sd=/var/spool/lpd/lp*: este é o diretório no qual vai ficar o arquivo de impressão, enquanto o *daemon* espera a impressora estar pronta para imprimir. Imagine a cena: vários usuários mandam imprimir diferentes arquivos e a impressora só pode imprimir um de cada vez. Enquanto

um dos arquivos é impresso os outros ficam esperando na fila, que costumamos chamar de *spool*. Quando a impressora termina de imprimir o primeiro trabalho, ela passa para o próximo da fila.

- *mx#0*: esse misterioso comando define o tamanho máximo do arquivo a ser impresso. Se não quiser impor nenhum limite, use o 0 (zero).
- *sh*: essa linha, que não é mais necessária no LPRng, avisa ao *daemon* que não deve ser impressa uma página de “rosto” com o nome de quem imprimiu as páginas seguintes, nome do arquivo e outras informações. O comportamento padrão do BSD LPD é imprimir essa folha de rosto, por isso a necessidade dessa opção. No LPRng o padrão é não imprimir.

Pronto, já temos a nossa primeira entrada de impressão. Agora devemos avisar ao *daemon* que nossa nova impressora existe. Primeiro, usamos o comando *checkpc*:

```
# checkpc -f
```

Este comando irá criar todos os arquivos de que nossa nova impressora precisa. Lembre de executá-lo sempre que adicionar uma nova impressora. Agora vamos notificar o *lpd*:

```
# lpc reread
```

Com esse comando avisamos ao *daemon* para ler novamente seus arquivos de configuração. Agora a nossa impressora já está pronta para uso.

Infelizmente nem tudo é simples como deveria ser. Para que funcione com apenas as definições acima, uma impressora deve entender arquivos texto no formato UNIX e arquivos PostScript. Normalmente elas apenas compreendem arquivos no formato DOS e usam um formato exótico qualquer para imprimir gráficos (como o PCL, da HP). Podemos resolver esse pequeno problema adicionando um filtro à nossa definição de impressora, como veremos mais adiante ainda neste artigo.

Imprimindo em rede...

Configurar uma impressora de rede é tão simples como uma impressora local; aliás, é até mais simples. De maneira

geral, para economizar trabalho, o filtro de impressão é instalado na máquina na qual está conectada a impressora, ou seja, não precisamos nos preocupar com ele. Uma entrada para uma impressora remota se parece com a seguinte:

```
lp1|remote-printer:\
:rp=lp:\
:rm=maquina-remota.lalala:\
:sd=/var/spool/lpd/lp1:\
:sh:
```

Bem semelhante à entrada anterior, não acham? A grande diferença são os campos *rp* e *rm*. O primeiro é o nome da impressora na máquina remota (no exemplo dado, *lp*) e o segundo é a máquina na qual a impressora está conectada (no exemplo, *maquina-remota.lalala*). Esta é a sintaxe tradicional do BSD LPD. No LPRng as coisas são ainda mais simples:

```
lp1|remote-printer:\
:lp=lp@maquina-remota.lalala:\
:sd=/var/spool/lpd/lp1:
```

O campo *:sh*: não aparece porque, como já foi dito, ele não é necessário no LPRng. Os campos *rm* e *rp* são mesclados em um único campo *lp*, onde informamos a impressora e a máquina à qual ela está conectada.

Com isso já sabemos configurar impressoras locais e impressoras remotas, bastam apenas algumas linhas no arquivo */etc/printcap*.

Uma fila para várias impressoras

Uma situação comum em ambientes corporativos é termos várias impressoras cumprindo a mesma função. Com isso, podemos ter uma impressora “mestre” que repassa os trabalhos às escravas à medida que elas forem desocupadas. Há duas vantagens nesse método:

1. Você vai usar melhor suas impressoras, já que não vai ficar com uma delas entupida de trabalho e outra parada.
2. Não precisa configurar *N* impressoras em todas as máquinas da sua rede. Configure só a impressora mestre e cuide das outras direto no servidor.

Montar uma configuração deste tipo usando o LPRng é fácil. Veja um exemplo a seguir:

```
master:\
:sd=/var/spool/lpd/%P:\
:sv=slave1,slave2,slave3:
```

Duas novidades, o uso do parâmetro `%P` e o campo `sv`. O `%P` apenas diz ao LPRng que o nome do spool é o mesmo da impressora (no nosso caso, `/var/spool/lpd/master`). Note que a cada exemplo vamos deixando isso cada vez mais simples.

Já a linha `sv` descreve quais são as impressoras “escravas”; `sv` significa *slave*. Chamei as impressoras escravas de `slaveN`, onde `N` é um número de 1 a 3. A configuração das impressoras escravas é idêntica à de uma impressora normal, exceto por um campo:

```
slave1:\
:lp=/dev/lp0:\
:sd=/var/spool/lpd/lp1:\
:mx#0:\
:ss=master:
```

Note a linha `ss=master`, que indica o nome da impressora mestre à qual essa escrava serve. Você deve incluir esse campo na configuração das impressoras escravas. Em seguida, execute os comandos `checkpc` e `lpc reread` para fazer as suas novas impressoras funcionarem.

Filtros...

Quando o assunto são os filtros, as coisas começam a ficar um pouco mais complicadas - e quase sempre precisamos lidar com eles. No caso de impressoras remotas, nem sempre isso é necessário. Impressoras de rede costumam compreender o formato PostScript e, no caso de impressoras conectadas a outra máquina *NIX, o filtro geralmente já está instalado na ponta “de lá”.

Mas existem as impressoras locais e nem sempre os filtros estão realmente instalados na máquina remota. É necessário um filtro diferente para cada tipo de impressora, às vezes até mais que um. Adicionar um filtro a uma de nossas entradas no `printcap` é fácil:

```
lp|printer1:\
:lp=/dev/lp0:\
:if=/caminho/para/o/filtro:\
:sd=/var/spool/lpd/lp:\
:mx#0:\
:sh:
```

Ao comparar com a entrada que temos na seção anterior, você verá que a única diferença é o campo `if`, onde está especificado o caminho completo para o filtro. O filtro é específico para um determinado modelo de impressora, no caso a Epson Stylus C42SX. O exemplo abaixo é voltado a ela:

```
#!/bin/sh
#
/usr/bin/gs -q -dSAFER -dBATCH ⤵
-dNOPAUSE -dPARANOIDSAFER ⤵
-sDEVICE=ijs ⤵
-sIjsServer=/usr/bin/ijsgimpprint ⤵
-dIjsUseOutputFD -sPAPERSIZE=a4 ⤵
-sDeviceManufacturer=EPSON ⤵
-sDeviceModel=escp2-c40sx ⤵
-sOutputFile=- -
```

O texto acima, que começa com `/usr/bin/gs` e termina com `-`, é escrito todo em uma única linha, sem quebras. Vamos dissecar esse monstro para ver como ele funciona:

- `/usr/bin/gs` é o GhostScript, nosso “superfiltro”. Ele já vem com vários filtros embutidos, além de permitir a adição de filtros novos através de módulos especiais.
- `-q -dBATCH -dNOPAUSE`: esses três comandos fazem com que o GhostScript não gere mensagens (modo silencioso, `-q`), que não haja uma pausa após a tradução de cada página (`-dNOPAUSE`) e que o prompt de comando do GhostScript não seja mostrado no final do trabalho (`-dBATCH`). Esses passos são importantes para que nada atrapalhe a impressão do nosso documento.
- `-dSAFER -dPARANOIDSAFER`: essas opções de segurança são utilizadas para garantir a integridade dos seus arquivos e evitar que algum código PostScript faça algo errado no seu sistema. Sim, PostScript é uma linguagem de programação e pode ser usada para fazer as coisas mais estranhas, como servidores Web.
- `-sDEVICE=ijs` escolhe o tipo de dispositivo a ser utilizado. Quando utilizamos o tipo `ijs`, preparamos o GhostScript para carregar um filtro externo. No exemplo, estamos usando os filtros do `gimp-print`.
- `-sPAPERSIZE=a4` informa o tamanho de papel. Usamos o formato A4.

- `-sOutputFile=-` configura o arquivo de saída para a saída padrão. O LPRng irá cuidar de receber a saída padrão e direcioná-la para a impressora.
- `-` simboliza a entrada padrão, que é por onde o nosso filtro de impressão está recebendo o arquivo.
- `-sIjsServer=/usr/bin/ijsgimpprint`: essa opção indica ao GhostScript quem será o responsável pelo filtro. Além desta, temos também outras opções específicas do servidor IJS.
- `-dIjsUseOutputFD` diz ao servidor IJS para ser bonzinho e obedecer aos nossos redirecionamentos para a saída padrão (normalmente ele se recusa a fazer isso).
- `-sDeviceManufacturer=EPSON -sDeviceModel=escp2-c40sx` especifica o fabricante e o modelo da impressora. É aqui que especificamos o filtro propriamente dito. Para ver uma lista com todas as impressoras suportadas pelo `gimp-print`, consulte a página de manual do `gimp-print-models`.

Vamos chamar este filtro de `myfilter`, transformá-lo em um arquivo executável (`chmod +x myfilter`) e colocá-lo no diretório `/usr/local/bin`. Nossa entrada de impressora vai ficar assim:

```
lp|printer1:\
:lp=/dev/lp0:\
:if=/usr/local/bin/myfilter:\
:sd=/var/spool/lpd/lp:\
:mx#0:\
:sh:
```

Infelizmente o trabalho ainda não acabou. Da maneira como fizemos, o filtro só consegue trabalhar com arquivos no formato PostScript. Embora a maior parte dos programas gráficos do mundo *NIX gere saída em PostScript, um filtro do mundo real tem que trabalhar com diversos formatos de arquivo, inclusive texto puro. Para isso, é necessário descobrir qual o tipo do arquivo que está sendo impresso.

Podemos fazer isso em nosso filtro, mas já existem diversos filtros prontos que incluem este e outros recursos. O melhor a fazer é utilizar algum programa auxiliar, como o `apsfilter`, que irá guiá-lo por uma série de menus em modo texto até que sua impressora esteja devidamente configurada. Se por acaso sua impressora não for suportada, ou se você

estiver planejando fazer algum uso mais “criativo” do filtro (como reduzir todas as páginas e imprimir duas por folha, etc...) já tem uma idéia do que fazer.

Controlando tudo isso: lp*

Depois de criar nosso filtro e configurar as filas de impressão, precisamos colocar tudo isso para funcionar. Afinal, não adianta nada termos impressoras configuradas e não imprimirmos nada. Os principais comandos usados para controlar o LPRng e enviar trabalhos para as impressoras são:

- *lpr* - Envia um trabalho (arquivo) para a impressora
- *lpq* e *lpstat* - Verifica o estado da fila de impressão
- *lprm* - Remove um trabalho da fila
- *lpc* - Controla o sistema de impressão

Vamos analisar cada um destes comandos, em detalhes, a seguir.

lpr

O comando mais importante, e com certeza o mais usado, é o *lpr*. Com ele podemos mandar um trabalho para uma impressora, seja ela local ou remota. A

forma de uso é bem simples. Para imprimir um arquivo digite:

```
# lpr arquivo
```

ou podemos enviar o arquivo a ser impresso para a fila via pipes.

```
# cat arquivo | lpr
```

Nos dois casos, o “arquivo” será enviado para o daemon de impressão e, no momento oportuno, para a impressora padrão. No BSD LPD, qualquer impressora com o nome *lp* será a impressora padrão. Já no LPRng as coisas são diferentes: a impressora padrão será a primeira entre as listadas no arquivo */etc/printcap*. Por pura tradição, geralmente o nome dela também é *lp*.

Existem outras maneiras de determinar a impressora padrão; uma das mais simples é colocá-la na variável de ambiente `PRINTER`. Assim, se quisermos que a impressora padrão seja a *lp_aux*, devemos digitar o comando:

```
# export PRINTER=lp_aux
```

Você pode colocar essa variável de ambiente no seu *.bashrc* ou *.bash_profile*. Também é possível especificar a impressora diretamente na linha de comando:

```
# lpr -Plp_aux arquivo
```

Ou, indo mais além, especificar uma impressora remota diretamente:

```
# lpr -Plp@outra_maquina arquivo
```

Além de enviar um trabalho para a impressora, o *lpr* possui diversas outras opções; veja algumas delas:

- *-C letra* determina a prioridade da impressão. A menor é *A* e a maior é *Z*. A configuração padrão envia todos os trabalhos como tendo a prioridade mais baixa. O *C* significa “class”.
- *-B* junta vários trabalhos em um só. Assim, você pode mandar imprimir vários arquivos como sendo apenas um trabalho. Útil para ajudar no gerenciamento dos trabalhos.
- *-Kn* imprimem cópias do trabalho enviado. É um recurso muito útil, porém pouco conhecido.

- *-b*: este parâmetro habilita o modo “binário”, no qual o sistema de impressão tenta manter o arquivo “intocado”. Muito útil quando utilizado em conjunto com uma impressora *raw* (sem filtro). Dessa maneira, o sistema de impressão se comporta praticamente como um *pipe*, enviando o trabalho diretamente da máquina cliente para a impressora.

lpq

Apesar dos nomes diferentes, *lpq* e *lpstat* são praticamente o mesmo comando, sendo o *lpstat* uma espécie de front-end para o *lpq*, com sintaxe e algumas opções diferentes. O objetivo desses comandos é verificar o estado de uma fila de impressão, se está ocupada, quem está imprimindo, se ela existe, entre outras informações. Vamos explicar como funciona o *lpq*, pois com ele é possível coletar todas as informações que desejamos. Caso precise de alguma informação adicional, consulte as páginas de manual dele e do *lpstat*. O comando básico é:

```
# lpq -Pall -s
```

Esse comando vai lhe mostrar todas as impressoras configuradas na máquina em que você está e quantos trabalhos estão na fila de cada uma delas. Uma saída típica seria:

```
# lpq -Pall -s
master@rachael 0 jobs (subservers lp)
lp@rachael 0 jobs (servng master)
mp3@rachael 0 jobs
lp1@rachael 0 jobs
```

O *-s* é de “short”; sem essa opção são apresentadas várias informações extras. O parâmetro *-P* funciona de maneira idêntica ao do comando *lpr*. A diferença aqui é o *all*, que é um apelido indicando todas as impressoras. Se quiser fazer a mesma pesquisa em uma máquina remota, basta digitar:

```
# lpq -Pall@maquinaremota -s
```

Com isso, mostramos as informações das filas de impressão configuradas na máquina remota.

Retirando o *-s*, temos várias outras informações sobre fila de impressão, sendo a principal a lista dos trabalhos sendo impressos. Uma saída típica inclui, entre outras informações, as mostradas no Quadro 1. Com isso podemos saber que:

1. Existe um trabalho sendo impresso (active), ele pertence ao usuário punk, é da classe A (a mais baixa), possui o número 187 e está sendo impresso desde as 20:35:53

2. O tal usuário punk também imprimiu um trabalho da mesma classe as 12:31:32. Diga-se de passagem, um arquivo bem maior, com quase 36k

Estas são informações bem interessantes; podemos obtê-las em várias impressoras ao mesmo tempo e, com isso, ter a qualquer momento um instante do que está acontecendo no sistema de impressão.

lprm

Esse é um comando simples de usar, mas que geralmente tem pouca utilidade. Normalmente, quando percebemos que estamos imprimindo algo errado, o trabalho já está no buffer da impressora e não mais na fila, portanto não há mais o que fazer. Mas enquanto ele estiver na fila (ou no caso de ainda não ter sido enviado totalmente para a impressora), pode ser removido com o comando *lprm*. A sintaxe padrão é:

```
# lprm
```

Isso irá remover da fila o último trabalho enviado. De maneira geral, um usuário comum pode remover apenas os trabalhos criados por ele. Se você quiser remover um trabalho específico da fila, pode fazer como em uma lanchonete e “pedir pelo número”.

```
# lprm numero_do_trabalho
```

E como descobrir o número do trabalho? Com o `<C>lpq<C>`, que o mostra na coluna `<I>Job<I>`. Dependendo do tamanho do trabalho, quando descobrimos qual o número ele já foi impresso. Para resolver este problema podemos usar:

```
# lprm -a
```

Que remove **todos** os trabalhos pertencentes à mesma fila do último trabalho enviado (*nunca* faça isso como root, a não ser que tenha um ótimo motivo). Da mesma maneira que nos outros comandos, a opção *-P* pode ser utilizada para especificar uma fila em especial ou todas, com a opção *all*. O usuário root (sempre ele) pode remover trabalhos de outros usuários:

```
# lprm -Plp nome_do_usuario
```

Esse comando remove o último trabalho enviado pelo usuário *nome_do_usuario* à fila *lp*. Como vimos, é um comando bem simples e fácil de ser utilizado. Todos os usuários deveriam conhecê-lo, o que poderia resultar na economia de algumas toneladas de papel.

lpc

Esse é o comando mais complexo do conjunto. Com ele controlamos todo o sistema de impressão, e podemos redirecionar trabalhos, verificar o estado das impressoras, reler os arquivos de configuração, ativar e desativar impressoras, entre vários outros truques que podem salvar nossos pescoços.

Um uso interessante é reposicionar os trabalhos na fila (útil para agradar os chefes). Tendo o número do trabalho de impressão em mãos (que pode ser conseguido com o comando *lpq*), digite:

```
# lpc topq nome_da_impresora > trabalho
```

Quadro 1: saída do comando *lpq*

```
# lpq -Pall@maquinaremota
Rank      Owner/ID          Class  Job   Files          Size  Time
active    punk@rachael +187  A      187   /etc/fstab      690   20:35:53
done      punk@rachael +592  A      592   /tmp/a2_kB6Sgp  37002  12:31:32
```

Com isso o trabalho será passado para o início da fila. Também é possível redirecionar trabalhos de uma impressora para outra. Se uma das suas impressoras parar de funcionar e houver uma longa fila de espera, basta digitar:

```
# lpc redirect impressora_quebrada nova_impressora
```

Quando consertar sua impressora, cancele o redirecionamento trocando o parâmetro *nova_impressora* por *off*. Se não houver outra impressora, você pode “estacionar” os trabalhos enquanto executa a manutenção:

```
# lpc holdall impressora
```

Isso irá segurar todos os trabalhos na fila. Depois de corrigir o problema, libere os trabalhos para serem impressos. Basta digitar o comando:

```
# lpc release impressora
```

É possível “segurar” e “liberar” trabalhos indicados por seus números. A sintaxe é: *lpc hold trabalho* e *lpc release trabalho*.

Outros parâmetros do LPC muito utilizados são:

- *reread*: Faz com que as configurações do servidor de impressão sejam lidas novamente. Seu uso é necessário quando os arquivos */etc/printcap* ou *lpd.conf* são alterados.
- *down impressora*: “Desliga” a impressora, fazendo com que ela pare de receber e de imprimir trabalhos
- *up impressora*: “Liga” a impressora, deixando-a pronta para receber trabalhos. Tanto neste como em vários outros comandos é possível substituir *impressora* por *all*, indicando todas as impressoras
- *status*: Mostra o estado atual das impressoras do sistema, se estão ativas, se estão redirecionadas, quantos trabalhos estão na fila etc.

Existem vários outros parâmetros que podem ser usados com o *lpc*, inclusive alguns muito úteis como o *stop*, *start*, *enable*, *disable*, *move* etc. Uma consulta à página de manual (*man lpc*) irá ajudá-lo a conhecer melhor essas opções.

Existem maneiras mais simples de configurar o LPRng (com o *apsfilter* [4], por exemplo). Mas a resolução de muitos

problemas, além de configurações mais exóticas, só pode ser conseguida com conhecimento detalhado de como as coisas funcionam, obtido através da análise dos arquivos de configuração. ■

INFORMAÇÕES

- [1] Página oficial:
<http://www.lprng.com>
- [2] Printing Cookbook:
<http://www.lprng.com/PrintingCookbook/index.html>
- [3] <http://www.piterpunk.hpg.ig.com.br/>
- [4] “Apsfilter, the number one magic filter”:
<http://www.apsfilter.org/>

SOBRE O AUTOR

Piter PUNK, também conhecido por alguns como Roberto Freires Batista, é usuário de Linux desde 1996, e trabalhou durante vários anos como administrador de sistemas. É o principal desenvolvedor do *slackpkg*, ferramenta para atualização de pacotes no Slackware. Escrever documentação e brincar com computadores MSX são dois de seus hobbies.



Remasterizando o Kurumin

Pinguim ao gosto do freguês

A parte mais interessante do Software Livre é que você não precisa se limitar a apenas usá-lo: pode colocar a mão na massa, corrigir problemas, adicionar novos recursos ou personalizar o que quiser. Veja aqui como criar sua própria distribuição a partir do Kurumin.

O Kurumin nasceu como uma personalização do Knoppix, que com o tempo ganhou vida própria. O Knoppix é uma distribuição baseada no Debian, que utiliza o módulo *cloop* para rodar a partir de uma imagem compactada de um sistema de arquivos gravada no CD-ROM. Além de rodar diretamente do CD-ROM, sem alterar nada no HD, ele inclui uma série de utilitários, com destaque para o *hwsetup*, que se encarrega de detectar todo o hardware da máquina durante o boot.

Mas a parte mais interessante do Knoppix é que você pode personalizar o CD, adicionando ou removendo programas, alterando as configurações ou o que mais for necessário para fazer uma distribuição adequada às suas necessidades. Além do Kurumin, existem hoje em dia algumas centenas de distribuições “filhas” baseadas no Knoppix, além de várias distribuições “netas”, desenvolvidas a partir do Kurumin ou de outras distribuições “filhas”. Isso tudo graças à simplicidade do processo de remasterização do CD, já dominado por muita gente. Algumas das vantagens do Knoppix:

- Ele já vem com um instalador, o que permite instalá-lo rapidamente no HD, mantendo toda a configuração de hardware feita durante o boot.
- É possível instalar qualquer um dos mais de 15.000 pacotes do Debian usando o *apt-get*.

- O conteúdo do CD é compactado, o que permite acomodar cerca de 2 GB de software num CD de 700 MB, mais do que suficiente para uma distribuição completa.

- É possível instalar drivers para SoftModems (WinModems) e outros tipos de hardware não suportado, programas comerciais e assim por diante. Você pode até mesmo usar o Wine para rodar alguns aplicativos Windows como o Microsoft Office.

As possibilidades são imensas. Você pode criar uma distribuição padrão para ser instalada em todos os PCs de sua empresa e ao mesmo tempo usá-la como forma de apresentar o Linux aos funcionários, enquanto mantém o Windows instalado no disco rígido. É possível criar CDs com softwares diversos para apresentá-los a seus clientes, e CDs para usos específicos, como discos de recuperação de sistema, documentação eletrônica e assim por diante.

Você pode começar a partir do Kurumin, a partir do Knoppix original ou a partir de qualquer uma das distribuições filhas ou netas. A escolha depende daquilo que você pretende fazer.



Montagem de Luciano Hagge sobre originais de "Bubbeis" e "Alpacá" - www.sx.chu

Como o Kurumin funciona

O CD do Kurumin contém apenas dois diretórios: */doc* e */KNOPPIX*, junto com alguns ícones e arquivos html. A pasta *doc* contém apenas os FAQs e manuais inclusos no sistema. A pasta *KNOPPIX* é a que nos interessa. Dentro dela você encontrará as imagens dos disquetes de boot e um arquivo que ocupa quase todo o espaço do CD.

Esse arquivo é nada mais do que uma imagem compactada da partição raiz (*/*) do sistema. O módulo *cloop* “engana” o Kernel, fazendo-o pensar que está acessando uma partição *ext2* no disco rígido. Quando é preciso abrir um arquivo, ele descompacta a porção correspondente da imagem, lê o arquivo e entrega os dados já “mastigados” para o Kernel.

Alguns diretórios que precisam de suporte a escrita, como `/home` e `/var`, são armazenados num ramdisk de 2 MB criado durante a inicialização. Este ramdisk pode crescer conforme necessário, desde que exista memória suficiente. Como nem todo mundo tem 256 MB, ou mais, de RAM, o sistema usa partições Linux swap, ou arquivos de troca (*pagefile.sys*) encontrados em partições Windows, como memória virtual.

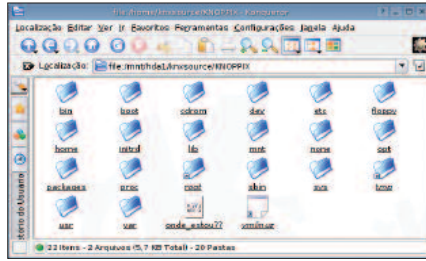
O módulo *loop* foi originalmente desenvolvido por Andrew Morton, que é atualmente o mantenedor do Kernel 2.6. Ironicamente, na época ele achou que o módulo não teria utilidade e o descartou. Algum tempo depois ele foi redescoberto pelo Klaus Knopper, que acabou por utilizá-lo como um dos componentes base do Knoppix. É um bom exemplo sobre como as coisas funcionam dentro do mundo Open Source.

Personalizando o CD

Para gerar uma versão personalizada do sistema precisamos descompactar essa imagem numa pasta do disco rígido, fazer as modificações desejadas, gerar uma nova imagem compactada e finalmente gerar a imagem de CD (arquivo .ISO) com nossa nova distro. Você vai precisar de:

- Uma partição Linux com cerca de 1.5 GB de espaço livre no disco (no caso do Kurumin) ou 3.5 GB caso esteja remasterizando o Knoppix ou outra distribuição que ocupe os 700 MB do CD. Calcule cerca de 2 GB de espaço para a imagem descompactada, 700 MB para a nova imagem e mais 700 MB para o novo arquivo ISO.
- Uma partição Linux Swap (ou um arquivo de troca de Windows) de 1 GB menos a quantidade de RAM do PC. Se você tem 256 MB de RAM, vai precisar de mais 768 MB de swap.

O sistema usa a memória RAM para armazenar a imagem compactada enquanto ela está sendo criada e só depois copia o arquivo para o HD. Uma imagem completa do Knoppix tem 700 MB, e ainda precisamos de memória para o uso normal do sistema. Note que a quantidade de memória varia de acordo com o tamanho da imagem gerada; você vai precisar de menos memória se estiver modificando uma distribuição menor, como o Kurumin.



O diretório *knxsource* irá conter todos os arquivos que compõem o sistema operacional.

Se você não tiver uma partição swap, pode criar um arquivo temporário usando o espaço livre da partição Linux. Para isso basta executar, como root, os comandos abaixo:

```
# dd if=/dev/zero >
of=/mnt/hda1/swap bs=1024 >
count=1000000
# mkswap /mnt/hda1/swap
# swapon /mnt/hda1/swap
```

Substitua o *hda1* pelo dispositivo correspondente à sua partição Linux, caso seja diferente. O Kurumin inclui um script que automatiza o processo de remasterização do CD, chamado *remasterizar-kurumin*, que pode ser encontrado no menu *Iniciar* ↙ *Sistema*.

Neste tutorial vou mostrar passo-a-passo como o processo funciona, explicando os comandos necessários para fazer tudo manualmente, de forma que você possa entender o que está acontecendo. O script serve para automatizar as partes chatas, permitindo que você se concentre na parte criativa da tarefa.

O *remasterizar-kurumin* possui 4 funções. As funções para extrair a imagem do Kurumin para a partição de remasterização e recriar a imagem do Kurumin (itens **a** e **d** no menu) devem ser executadas com o Kurumin rodando do CD. Elas se destinam justamente a extrair os arquivos que compõem o sistema operacional para a partição de trabalho.

As funções *abrir o chroot da imagem* e *gerar o novo ISO* (itens **B** e **C**) podem ser executadas tanto do CD quanto com o Kurumin instalado no disco rígido. Elas se destinam a fazer alterações no sistema e depois gerar a nova imagem ISO com as alterações.

Uma observação importante é que é preciso usar a mesma versão do módulo *loop* instalada no sistema de desenvolvimento para fechar o arquivo compac-

tado. Em outras palavras, se você está remasterizando o Kurumin com uma versão personalizada do kernel 2.6.7, por exemplo, essa mesma versão do kernel deve estar sendo executada no Kurumin instalado no disco rígido ao fechar a imagem do sistema. Se você executar versões diferentes do kernel, ou estiver usando outra distribuição, as versões do *loop* serão diferentes e o novo CD simplesmente não vai “dar boot”.

Você pode criar as partições necessárias usando um CD de instalação do Mandrake ou o utilitário *cfdisk* incluso no próprio Kurumin. Pode também aproveitar uma partição de uma distribuição Linux já existente no disco rígido, desde que ela possua espaço livre suficiente. Lembre-se de que nesse caso não é possível utilizar uma partição Windows; deixe-a bem quietinha em um canto.

Comece inicializando o sistema a partir de um CD do Kurumin. Abra um terminal e rode o comando *sudo passwd* para definir a senha de root. Logue-se como root e monte a sua partição de trabalho. Um detalhe importante é que você deve montar a partição dentro de um terminal usando o comando *mount* e não usando os atalhos no desktop. Eles montam as partições adicionando o parâmetro *nODEV*, que impede que os scripts direcionem suas saídas para o dispositivo */dev/null*, o que causa uma série de erros. Digite qualquer um dos dois comandos a seguir:

```
# mount /dev/hda1 /mnt/hda1
# mount -t reiserfs /dev/hda1 >
/mnt/hda1
```

No segundo comando, o parâmetro *-t* indica que a partição usa o sistema de arquivos *ReiserFS*. Substitua o *hda1* pelo dispositivo correspondente à sua partição de trabalho.

Agora você deve criar duas pastas, uma para abrigar a imagem descompactada e outra para guardar os arquivos que irão no CD, fora da imagem. O que estiver dentro da imagem compactada só será acessível após inicializar o sistema pelo CD, mas os arquivos fora da imagem podem ser acessados a partir de qualquer sistema operacional. Por pura tradição, vamos usar os mesmos nomes de pasta sugeridos no tutorial *Knoppix Remastering How-To*:

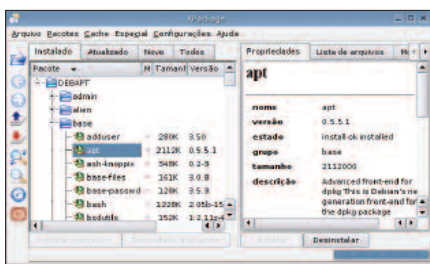
```
# mkdir /mnt/hda1/knxmaster
# mkdir /mnt/hda1/knxsource
# mkdir /mnt/hda1/knxsource2
/KNOPPIX
```

Em seguida copie todo o conteúdo do CD-ROM para a pasta knxmaster. Não é necessário copiar o arquivo compactado; ele será substituído mais adiante.

```
# cp -a /mnt/cdrom/*
/mnt/hda1/knxmaster
```

Vamos entender o que vai dentro do CD:

O arquivo *index.html* encontrado no raiz do CD é a página web que é aberta automaticamente quando o ambiente gráfico é carregado. Você pode, por exemplo, incluir uma cópia off-line do seu site no CD. Os arquivos *autorun.bat* e *autorun.inf* são responsáveis pela auto-execução do CD no Windows, que também abre a mesma página HTML. Você pode editar estes arquivos para que seja aberta uma página diferente ou mesmo um programa que mostre uma tela de apresentação mais elaborada.



O KPackage pode ser usado para instalar e remover pacotes no sistema.

Dentro da pasta *KNOPPIX* você encontrará o arquivo *background.gif* que é o papel de parede usado no KDE. Também é possível substituí-lo por outra imagem de seu agrado. Os arquivos *boot.img* e *boot.cat* são as imagens do disquete de boot, necessárias para tornar o CD “bootável”. Não as apague!

Veja que esses arquivos estão “fora” da imagem compactada, ou seja, eles podem ser lidos em qualquer sistema operacional, mesmo por alguém que usa o Windows e colocou o CD no drive por acaso. A forma mais simples de personalização do sistema é simplesmente substituir esses arquivos para criar um CD do Knoppix levemente personalizado.

Em um CD do Kurumin há cerca de 500 MB de espaço livre para você colocar

o que quiser. Você pode fazer um CD com utilitários de recuperação do sistema, livros ou outros tipos de documentação e muito mais. Uma revista pode usar esse recurso para lançar CDs “híbridos”, que são discos bootáveis com o Kurumin e ao mesmo tempo contêm o material normal da edição.

Neste caso só é preciso gerar outra imagem ISO bootável. Basta colocar os arquivos da pasta *knxmaster* no raiz do CD e indicar o arquivo *boot.img* como imagem de boot. Caso o programa peça por um catálogo de boot, indique o arquivo *boot.cat*.

No *X-CD-Roast* selecione a opção *Master Tracks*, indique a pasta onde estão os arquivos do CD (opção *Master Source*) e em *Boot Options* indique os arquivos *boot.img* e *boot.cat*. Se você prefere a linha de comando, pode usar o próprio Kurumin para gerar a nova imagem. Em um terminal, vá até a pasta *knxmaster* e use o comando:

```
$ mkisofs -pad -l -r -J -v -V
“KNOPPIX” -b KNOPPIX/boot.img -c
KNOPPIX/boot.cat -hide-rr-moved
-o /algum_lugar/knoppix.iso
/algum_lugar/knxmaster
```

Substitua o */algum_lugar/knoppix.iso* pela pasta onde deseja colocar a imagem ISO e o */algum_lugar/knxmaster* pelo caminho completo até a pasta *knxmaster* no seu HD, como em:

```
$ mkisofs -pad -l -r -J -v -V
“KNOPPIX” -b
/mnt/hda1/KNOPPIX/boot.img -c
/mnt/hda1/KNOPPIX/boot.cat
-hide-rr-moved -o
/mnt/hda1/knoppix.iso
/mnt/hda1/knxmaster
```

Isso também pode ser feito no Windows. No *Easy CD Creator*, por exemplo, clique em *Arquivo* > *Novo Projeto de CD* > *CD de inicialização*. Na tela que surgirá escolha *Emulação por disquete (1.44 MB)*, *Usar um arquivo de imagem existente* e aponte para o arquivo *boot.iso* na pasta do Knoppix.

Se você estiver usando o utilitário *remasterizar-kurumin*, dê boot pelo CD do Kurumin, use a opção *A* para extrair os arquivos do CD, jogue os demais arquivos que serão incluídos no CD den-

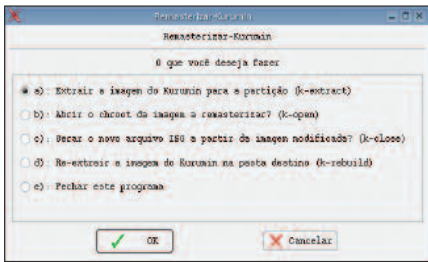
tro da pasta *knxmaster* e use a opção *C* para gerar o novo ISO, sem passar pela opção *B*, que permite editar os arquivos dentro da imagem compactada.

Criando seu sistema de desenvolvimento

Vamos ao que interessa, que é alterar o sistema em si. O primeiro passo é descompactar a imagem dentro da pasta *knxsource/KNOPPIX*, criada anteriormente. Mas atenção! para que o comando a seguir funcione você deve inicializar seu computador a partir do CD do Kurumin, já que sua função é copiar o sistema de arquivos montado durante o boot para a pasta indicada:

Listagem 1 - Instalando a máquina virtual Java

```
01 #!/bin/sh
02 cd /usr/lib
03 wget -c
ftp://ftp.funet.fi/pub/Linux/
java/jdk/blackdown.org/JDK-
1.4.1/i386/01/j2re-1.4.1-01-
linux-i586-gcc3.2.bin
04 wget -c
ftp://ftp.gwdg.de/pub/languages/
/java/linux/JDK-1.4.1
/i386/01/j2re-1.4.1-01-linux-
i586-gcc3.2.bin
05 chmod +x j2re-1.4.1-01-linux-
i586-gcc3.2.bin
06 ./j2re-1.4.1-01-linux-i586-
gcc3.2.bin
07 ln -sf
/usr/lib/j2re1.4.1/plugin/i386
/mozilla/javaplugin_oji.so
/usr/lib/mozilla/plugins/
javaplugin_oji.so
08 ln -sf
/usr/lib/j2re1.4.1/plugin/i386
/mozilla/javaplugin_oji.so
/usr/lib/MozillaFirebird/
plugins/javaplugin_oji.so
09 ln -sf
/usr/lib/j2re1.4.1/plugin/i386
/mozilla/javaplugin_oji.so
/usr/lib/firefox/plugins/
javaplugin_oji.so
10 ln -sf
/usr/lib/j2re1.4.1/bin/java
/usr/bin/java
11 rm -f j2re-1.4.1-01-linux-
i586-gcc3.2.bin
```

O script *remasterizar-kurumin* automatiza a maior parte do trabalho.

```
# cp -Rp /KNOPPIX/* \
/mnt/hda1/knxsource/KNOPPIX
```

Sempre lembrando que você deve substituir *hda1* pela partição de trabalho que estiver usando. Essa etapa demora um pouco, cerca de 10 minutos numa máquina com um processador Intel Celeron 600 MHz com um CD-ROM de 40x. Quando terminar você verá que a árvore de diretórios do sistema está acessível dentro da pasta */mnt/hda1/knxsource/KNOPPIX*. Caso não queira fazer o processo “na mão”, você pode usar a opção A do utilitário *remasterizar-kurumin*.

Você deve estar se perguntando se o próximo passo é acessar a pasta e sair editando os arquivos de configuração e instalando coisas manualmente. Bem, isso até seria possível para alguém sem muito o que fazer, mas existe uma forma muito mais fácil de trabalhar dentro da pasta de desenvolvimento, utilizando o comando *chroot*. Ele permite transformar nossa pasta de trabalho no diretório raiz do sistema, de modo que você pode adicionar e remover pacotes e até mesmo abrir o KDE e alterar suas configurações. Tudo o que você fizer dentro da janela do *chroot* alterará seu novo CD bootável. Para ativá-lo, use o comando:

```
# chroot /mnt/hda1/knxsource/\
KNOPPIX
```

Antes de começar a trabalhar, monte o diretório */proc* dentro do ambiente *chroot*. Sem isso, você não terá acesso a muitos recursos:

```
# mount -t proc /proc proc
```

Fazendo isso você já terá acesso completo via linha de comando. O próximo passo é iniciar o modo gráfico para trabalhar mais confortavelmente. Isto pode ser feito usando o *Xnest*, um servidor X

que permite que você abra o KDE do seu sistema de desenvolvimento dentro de uma janela no servidor X principal. Abra mais um terminal dentro do seu sistema host e use os comandos:

```
$ xhost +
```

Isto desativa o controle de acesso do X, o que previne alguns problemas.

```
$ Xnest :1
```

Isso abre uma janela com o segundo servidor X, mostrando uma tela cinza com o cursor do mouse. Volte para o terminal do ambiente *chroot* e use o comando:

```
# k-home1
```

Esse script copia o diretório */etc/skel*, onde ficam as configurações padrão do sistema, para a pasta */home/kurumin* e usa o utilitário *su* para “se logar” como o usuário *kurumin*. Execute o segundo script, que abrirá o KDE do sistema de desenvolvimento dentro da janela do *Xnest*, permitindo que você trabalhe com todos os aplicativos gráficos:

```
$ k-home2
```

Para acessar a internet de dentro do ambiente *chroot* você deve editar o arquivo */etc/resolv.conf*, adicionando o endereço do servidor DNS de seu provedor de acesso. Veja um exemplo:

```
search home
nameserver 200.177.250.10
nameserver 200.176.2.10
```

É agora que começa nosso trabalho de personalização propriamente dito. Como já disse, o Knoppix é baseado no Debian, o que permite que você instale qualquer pacote *.deb*, binários diversos e compile programas a partir de seu código fonte. Você pode ainda utilizar os ícones mágicos e os scripts de configuração do Kurumin, com poucas limitações.

Um detalhe muito importante: Quando terminar de trabalhar dentro da janela do *Xnest*, feche o KDE, volte para o terminal do ambiente *chroot* e pressione *Ctrl+D* para encerrar a sessão do usuário Kurumin e voltar a ser *root*. A seguir, execute o terceiro script:

```
# k-home3
```

Ele salva no diretório */etc/skel* todas as alterações feitas, finalizando o trabalho.

Lembre-se que dentro do ambiente *chroot* você pode criar novos usuários usando o comando *adduser*, como em *adduser jose*. Para “se logar” como o usuário criado basta usar um *su jose* seguido da senha (que pode ser definida ou alterada com o comando *passwd jose*) e depois digitar o comando *exit* para voltar a ser *root*.

Adicionando e removendo programas

A instalação de novos programas dentro da janela do ambiente *chroot* ou através do KDE rodando dentro do *Xnest* é feita da mesma forma que num sistema Kurumin ou Debian no disco rígido. Existem algumas peculiaridades quando o sistema roda a partir do CD, como o fato de que apenas o */home* e alguns arquivos de configuração possuem permissão de escrita (já que no geral o CD é “read-only”), por isso eventualmente alguns programas podem não funcionar.

Por exemplo, rodar programas Windows dentro do Wine pode ser um problema, já que os arquivos dentro da pasta *.wine/fake_windows* (o *C:* para os

Listagem 2 - Instalando o OpenOffice.org

```
#apt-get install prelink
#apt-get install openoffice.org
#apt-get install openoffice.org-bin
#apt-get install openoffice.org-2
help-en
#apt-get install openoffice.org-2
debian-files
#apt-get install openoffice.org-2
110n-pt-br
#apt-get install openoffice.org-2
mime1nk
#apt-get install openoffice.org-2
crashrep
#apt-get install oooqs-kde
#apt-get install mspell-pt-br
#apt-get install libming-fonts-2
openoffice
#apt-get install openoffice.org-2
thesaurus-pt-br
#apt-get install openoffice.org-2
hyphenation-pt-br
```

Listagem 3 - Instalando o Adobe Acrobat Reader

```

01 #!/bin/sh
02 cd /packages
03 mkdir acrobat
04 cd acrobat/
05 wget -c
    http://ardownload.adobe.com/
    pub/adobe/acrobatreader/unix/
    5.x/linux-508.tar.gz
06 tar -zxvf linux-508.tar.gz
07 sudo ./INSTALL
08 cd ..
09 rm -rf acrobat/
10 ln -s
    /usr/local/Acrobat5/bin/
    acroread /usr/bin/acroread
11 ln -s
    /usr/local/Acrobat5/bin/
    acroread /usr/bin/acrobat
12 cat <<EOF
    >/usr/share/applnk/Office/
    adobe.desktop
13 [Desktop Entry]
14 Encoding=UTF-8
15 Name=Acrobat Reader 5
16 Exec=/usr/local/Acrobat5/bin/
    acroread
17 X-KDE-Library=libkwordpart
18 GenericName=
19 MimeType=
20 Type=Application
21 Icon=acroread
22 EOF

```

programas Windows), precisam de permissão de escrita. No caso de programas pequenos, é possível copiar a pasta para o ramdisk, mas se você quiser rodar um programa grande, como o Office ou o Photoshop, a única forma de rodar a partir do CD seria copiar a pasta *.wine/fake_windows* para uma partição do disco rígido ou um diretório compartilhado na rede, já que ela ficaria grande demais para o ramdisk.

Para instalar programas usando o apt-get, use os comandos de sempre:

```
# apt-get update
```

Isto irá baixar e atualizar a lista dos pacotes disponíveis no repositório apt. Em seguida digite:

```
# apt-get install programa
```

para instalar o programa desejado.

No caso de programas com instalador próprio, como a máquina virtual Java, copie o pacote de instalação para dentro da pasta *knxsource/KNOPPIX* e o execute dentro do ambiente chroot. O script na Listagem 1 baixa e instala a máquina virtual Java da Blackdown.org.

Se você quiser gerar uma imagem com o OpenOffice pré-instalado, prefira os pacotes disponíveis no repositório apt-get; eles funcionam melhor a partir do CD do que o pacote tradicional disponí-

vel no site OpenOffice.org. Veja na Listagem 2 os comandos necessários para instalá-lo, já com o corretor ortográfico e dicionários em português.

A Listagem 3 mostra um script que instala o Acrobat Reader e cria o respectivo ícone no menu “iniciar”:

Para remover programas que você não vai usar, use o comando *apt-get remove programa*. Não se esqueça de, antes, executar *apt-get update*, para que ele tenha a oportunidade de checar as dependências. Você pode ver uma lista de todos os pacotes instalados, junto com a descrição, versão e tamanho de cada um com o *kpackage*, o gerenciador de pacotes do KDE. É possível inclusive fazer a instalação e remoção diretamente através dele.

Preste atenção antes de sair removendo as coisas! Muitos pacotes são “âncoras”, necessários para outros pacotes. Se você tentar remover o *kdelibs4*, uma das bibliotecas centrais do KDE, vai precisar remover junto quase todos os programas relacionados a ele. Não existem muitos pacotes “desnecessários” no Kurumin. A única forma de diminuir o tamanho do sistema é realmente removendo programas. Ao trabalhar sobre o Knoppix ou outra distribuição maior, dá para liberar um bom espaço removendo os pacotes de internacionalização do KDE, as pastas de documentação em */usr/share/doc* e programas especializados, usados por poucos usuários finais, como o *emacs* e o *Vim*.

Gerando a nova imagem

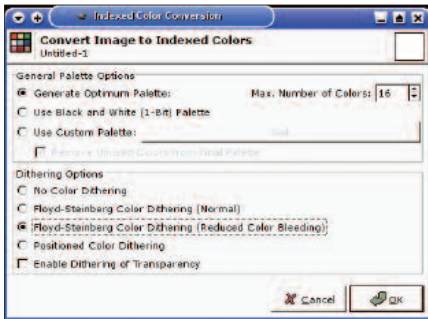
Ao remover muitos pacotes do sistema é normal que surjam pacotes “órfãos”, bibliotecas e arquivos que não são mais usados por nenhum programa e servem apenas para ocupar espaço. Você pode obter uma lista dessas “sobras” facilmente usando o comando *deborphan* e remover com segurança os pacotes listados usando o apt-get. Isso quase sempre vai economizar mais alguns megabytes. Se você usou o apt para instalar pacotes, pode liberar mais espaço limpando o cache local de pacotes. O Kurumin inclui um script que automatiza a limpeza deste cache, removendo também as listas de pacotes disponíveis nos repositórios. Basta digitar: *k-clean-apt-get*. Apague também o histórico de comandos do usuário *root*. Esse arquivo armazena os últimos comandos digitados, e



A imagem exibida antes da inicialização do sistema pode ser facilmente modificada

não existe necessidade de divulgá-los ao mundo. Aproveite para eliminar também o diretório `.rr_moved`.

```
# rm -f /home/root/.bash_history
# rm -rf /.rr_moved
```



Ao mudar a imagem de boot, certifique-se de que ela esteja no formato correto.

Finalmente chegou hora de deixar o ambiente chroot e gerar a nova imagem de sistema. Desmonte o diretório `/proc`:

```
# umount /proc
```

E tecla `CTRL + D` para fechar o ambiente chroot. O próximo passo é gerar a nova imagem compactada do sistema. Isso pode demorar um pouco, o tempo necessário diminui proporcionalmente conforme o tamanho da imagem. Essa etapa é automatizada pela função `C` do utilitário `remasterizar-kurumin`.

Antes de tentar gerar a imagem, use o comando `free` para verificar se a memória swap está ativada. Se necessário, formate novamente a partição swap e reative-a com os comandos `mkswap /dev/hda2` e `swapon /dev/hda2`. Sempre substituindo `hda2` pelo dispositivo correspondente à sua partição de swap. O comando para gerar a imagem é:

```
# mkisofs -R -V "Meu_CD" \
  -hide-rr-moved -pad \
  /mnt/hda1/knxsource/KNOPPIX | \
  /usr/bin/create_compressed_fs \
  -65536 > \
  /mnt/hda1/knxmaster/KNOPPIX/KNOPPIX
```

Não é necessário entender o comando anterior para usá-lo, mas vamos ver, em detalhes o que faz cada uma das opções:

- **mkisofs**: Este é o programa mais usado para gerar imagens ISO no Linux, inclusive por programas gráficos como o X-CD-Roast.

- **-R**: Ativa as extensões Rock-Ridge, que adicionam suporte a nomes de arquivos longos no Linux.

- **-V "KNOPPIX"**: O nome do volume. Você pode substituir `Meu_CD` por qualquer outro nome.

- **-hide-rr-moved**: Esconde o diretório `RR_MOVED` caso ele exista. Apenas uma questão estética.

- **-pad**: Para prevenir problemas de leitura, o tamanho total da imagem deve ser sempre um múltiplo de 32 KB. Este parâmetro verifica isso e adiciona alguns bits zero ao final da imagem para "arredondar" o arquivo.

- **/mnt/hda1/knxsource/KNOPPIX**: Este é o diretório fonte, onde está a imagem descompactada do sistema. Não se esqueça de substituir `hda1` pelo dispositivo correspondente à partição correta em sua máquina.

- **| /usr/bin/create_compressed_fs - 65536**: Este é o grande truque. Ele direciona toda a saída do comando para o programa `create_compressed_fs`, um utilitário incluso no CD, que se encarrega de compactar os dados. Note que, por causa do uso deste comando, você só poderá gerar a imagem compactada a partir do Kurumin ou outra distro baseada no Knoppix.

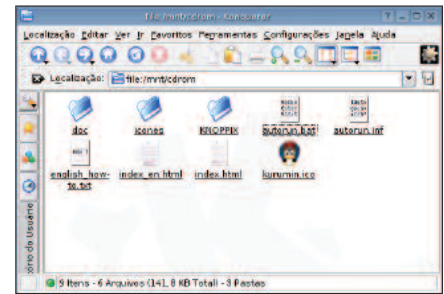
- **/mnt/hda1/knxmaster/KNOPPIX/KNOPPIX**: Este é o diretório de destino, onde será gravada a imagem compactada. Novamente, substitua `hda1` pelo nome da partição correta em sua máquina.

É provável que seu micro fique um pouco lento durante e logo após a geração da imagem, já que o processo consome quase toda a memória RAM disponível. Não se assuste, tudo logo volta ao normal. Agora falta apenas fechar a imagem do CD. Digite os comandos abaixo:

```
# cd /mnt/hda1/knxmaster
# mkisofs -pad -l -r -J -v -V \
  "Meu_CD" -b KNOPPIX/boot.img -c \
  KNOPPIX/boot.cat -hide-rr-moved \
  -o /mnt/hda1/kurumin.iso \
  /mnt/hda2/knxmaster
```

`/mnt/hda1/kurumin.iso` é a localização e nome da imagem ISO que será criada.

Ao contrário da geração da imagem compactada, a geração da imagem ISO é bastante rápida, depende apenas da



O Conteúdo do CD do Kurumin.

velocidade do seu HD. Assim que o processo for concluído você pode gravar seu novo CD e ver se tudo está funcionando como desejado.

Recomendo que você feche a imagem e grave um novo CD periodicamente, pois além de ajudar a detectar problemas, os CDs gravados servem como um ponto de recuperação. Caso mais adiante você faça alguma alteração que quebre o sistema e você não saiba como reverter, basta inicializar o computador com o último CD gerado, apagar o conteúdo da pasta `/knxsource/KNOPPIX` e extrair novamente a imagem. Você terá seu sistema de volta da forma como estava quando gravou o CD. Este processo corresponde à função **D** do utilitário `remasterizar-kurumin`.

Outra alternativa, caso você queira economizar discos CD-R, é arquivar as imagens ISO numa pasta do disco rígido e testá-las conforme necessário em uma máquina virtual, como o VMWare ou o QEMU, incluso no Kurumin com o nome de Kurumin-Emu.

No próximo artigo veremos como personalizar o KDE e fazer algumas modificações avançadas, como a alteração dos scripts de inicialização, do usuário padrão e como modificar o script responsável pela instalação do sistema no disco rígido. Até lá!

INFORMAÇÕES

- [1] Kurumin: <http://www.guiadohardware.info/kurumin/>
- [2] Site do Autor: <http://www.guiadohardware.net>
- [3] Linux-Live: <http://www.linux-live.org/>
- [4] Knoppix: <http://www.knoppix.org>
- [5] Portal Knoppix: <http://www.knoppix.net>
- [6] QEMU: <http://fabrice.bellard.free.fr/qemu/>
- [7] VMWare: <http://www.vmware.com>

Bits acelerados com o BitTorrent

Carga distribuída

Quanto mais popular um software, mais largura de banda é necessária para distribuí-lo. Resolva este dilema sem gastar fortunas com novos links.

POR CARSTEN SCHNOBER

Popularidade é bom, mas vem às custas da necessidade de uma imensa largura de banda para atender às hordas de usuários vorazes que querem baixar a versão mais recente de um software a partir dos servidores oficiais. A transferência de dados pode se tornar um grande problema, caso as empresas não consigam distribuir a carga.

Usuários que queiram baixar um arquivo grande, como uma imagem de CD de uma distribuição Linux, usando uma conexão discada, vão precisar de muita paciência caso o download seja interrompido no meio do caminho. Desenvolvedores realmente interessados em distribuir seu software à comunidade precisam encontrar meios de evitar esse problema.

A teoria...

O BitTorrent (literalmente “Torrente de Bits”) é um conceito que pode ajudar ambos os lados. Pessoas que precisam hospedar grandes quantidades de dados - como uma ou mais imagens de CD - são os “semeadores” (seeders), no jargão do BitTorrent, responsáveis pela distribuição inicial do arquivo. Os arquivos são divididos em partes menores e, após baixar a primeira parte, um cliente BitTorrent automaticamente se reconfigura como um “parceiro” (peer) e oferece esta mesma parte às outras máquinas na rede. É mais ou menos o comportamento



de um **servidor espelho** (mirror server), com a diferença de que o cliente só tem um pedaço do arquivo completo.

O próximo usuário que quiser baixar este arquivo pode encontrá-lo em qualquer uma das fontes acima - o semeador ou o parceiro - automaticamente tornando-se um parceiro. Após receber todas as partes do arquivo, e desde que você não feche o cliente BitTorrent, sua máquina automaticamente se transforma em um servidor, oferecendo o arquivo completo aos outros membros da rede, sem precisar baixar mais nada. Em outras palavras, sua máquina também se torna uma “semeadora”.

Da forma como foi projetado, o sistema permite que a largura total de banda seja dividida entre o grupo de usuários. Quanto mais pessoas estiverem interessadas em um arquivo, mais parceiros existirão. E quando um grupo suficientemente grande de parceiros e

semeadores se formar, o distribuidor original do arquivo não precisará mais se preocupar com o consumo de banda de sua conexão e os usuários estarão felizes com a alta velocidade de download. O número de parceiros é proporcional ao interesse gerado pelo arquivo que está sendo oferecido; dessa forma a fonte primária não deve ter dificuldades com a rede (como sobrecarga), mesmo que o número de parceiros seja pequeno.

Um “rastreador” (tracker) coordena todo este processo. Um rastreador é um servidor que conversa com os parceiros, perguntando a eles quais pacotes já foram baixados e podem ser oferecidos na rede. Quando um novo parceiro se conecta a um rastreador, recebe uma lista de parceiros ativos e semeadores, de onde pode começar a baixar o arquivo.

Para ajudar a encontrar o rastreador apropriado, há um arquivo .torrent para cada arquivo (ou conjunto de arquivos)

GLOSSÁRIO

Servidor espelho: Para ajudar a aliviar um pouco a carga de um servidor, espelhos oferecem o mesmo conteúdo (arquivos) em locais diferentes.

Hash: Uma seqüência numérica gerada a partir de um arquivo por um algoritmo matemá-

tico não-reversível. A mesma entrada (arquivo) vai produzir sempre o mesmo hash; entradas diferentes produzirão hashes diferentes. Ou seja, o hash serve como um identificador único para o arquivo, já que não há dois arquivos com hashes iguais.

Python: Linguagem de programação para Linux e sistemas Unix, além do Windows, OS/2 e Mac OS. Por ser interpretada, não é necessário um compilador para transformar o programa em código de máquina, apenas do interpretador Python para executá-lo.

oferecido na rede, contendo informações cruciais como o endereço IP do rastreador, a lista dos arquivos oferecidos e seus respectivos tamanhos e um código "Hash" único. Um rastreador não baixa nenhum arquivo e, portanto, consome muito pouca banda; só é necessária uma conexão permanente à Internet.

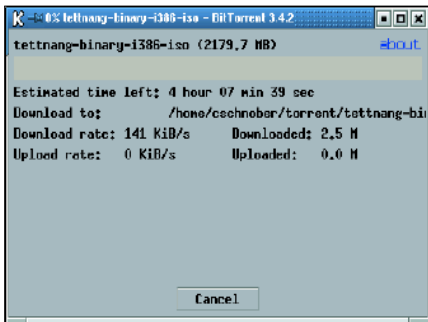


Figura 1: O BitTorrent permite downloads em alta velocidade.

...e a prática

Um usuário precisa do .torrent correspondente ao arquivo que deseja baixar antes de iniciar o download. Esses arquivos estão disponíveis em servidores espalhados pela Internet e podem ser facilmente encontrados com mecanismos de busca ou através de links em sites especializados. Supondo que você queira os CDs da distribuição Linux Fedora, basta digitar *Fedora* e *Torrent* em seu site de busca favorito para ser rapidamente levado ao endereço correto. Além disso, a página oficial do BitTorrent [1] traz alguns links.

A versão "Sarge" da distribuição Linux Debian contém um pacote com um cliente BitTorrent pronto para usar, que pode ser instalado com o comando `apt-get install bittorrent`. Se sua distribuição não contém um pacote do BitTorrent, baixe o código-fonte do programa no site oficial e descompacte-o com o comando `tar -zxvf BitTorrent-3.4.2.tar.gz`.

O cliente BitTorrent oficial é escrito na linguagem de programação **Python**, e precisa de alguns pacotes adicionais para funcionar. Além do próprio Python, presente em todas as distribuições Linux, o cliente gráfico precisa do conjunto de "widgets" (elementos de uma interface gráfica) `wxWidgets`. No Suse Linux este pacote é chamado de `python-wxGTK` e no Mandrake é chamado `wxPythonGTK`.

Salve o arquivo .torrent em seu disco rígido e execute o cliente BitTorrent gráfico,

encontrado no diretório BitTorrent-3.4.2 criado durante a descompactação do código-fonte, para iniciar o download:

```
python btdownloadgui.py ⤴
tettngang-binary-i386-iso.torrent
```

Este exemplo inicia o download dos CDs da distribuição Linux Fedora Core 2 (codinome "Tettngang"). Usuários do Debian podem usar o comando a seguir:

```
btdownloadgui tettngang-binary-⤴
i386-iso.torrent
```

Indique o diretório onde deseja colocar o arquivo baixado (veja a Figura 1).

Além do cliente gráfico, o BitTorrent também tem um utilitário para download em modo texto. Execute o programa `btdownloadcurses.py` em vez de `btdownloadgui.py` para uma interface com janelas em modo texto (usando a biblioteca `ncurses`), ou `btdownloadheadless.py` para a versão em linha de comando.

O comando a seguir executa múltiplos downloads em paralelo:

```
python btlaunchmanycurses.py ⤴
~/torrents
```

Este exemplo lê todos os arquivos .torrent no diretório `~/torrents/` e inicia todos os downloads em uma única janela em modo texto (veja a Figura 2). O cliente gráfico não possui suporte a múltiplos downloads simultâneos, mas você pode contornar o problema executando múltiplos clientes ao mesmo tempo.

A ferramenta `bbtshowmetainfo.py` lhe mostra informações sobre o conteúdo de uma torrent. Ao executar o script, especifique como parâmetro o nome de um arquivo .torrent para ver a lista e tamanho dos arquivos nele incluídos.

```
python bbtshowmetainfo.py ⤴
tettngang-binary-i386-iso.torrent
```

Mercado de Trocas

As torrents se tornaram tão populares que praticamente todo pedido de download encontra ao menos um parceiro ou semeador livre. Contudo, se um parceiro recebe múltiplos pedidos de download ao mesmo tempo, ele precisa de um meio para determinar qual deles terá prioridade. O parceiro irá sempre esco-

lher o candidato que tem mais partes a oferecer em troca. Portanto, a primeira parte de um arquivo pode demorar um pouco a chegar, mas a taxa de download vai aumentar a cada parte extra que você ajudar a colocar na rede.

Esta técnica impede que os clientes baixem arquivos sem dar nada em troca. Essa é uma boa idéia, já que tal comportamento "predatório" impediria o equilíbrio de carga (*loadbalance*). As coisas ficam mais complexas quando você tenta se conectar através de um roteador ou firewall que bloqueie seu contato com o mundo exterior. Você fica impedido de oferecer partes do arquivo aos outros parceiros - isso significa menores velocidades de download, já que haverá mais pedidos que ofertas.

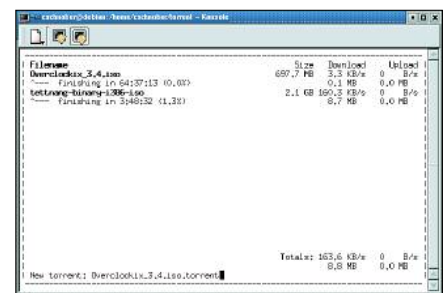


Figura 2: Múltiplos downloads em paralelo em uma janela em modo texto.

Se você puder alterar as configurações de seu firewall, pode resolver o problema permitindo conexões vindas da Internet nas portas TCP 6881 a 6889. De outra forma, parceiros que suportam acesso externo não tomarão conhecimento de seus colegas "ocultos" até que recebam um pedido deles, já que os parceiros anunciam automaticamente o que têm a oferecer quando conversam. Ou seja, clientes atrás de um firewall ainda vão receber uma resposta positiva aos seus pedidos de download, embora sejam menos conhecidos na rede e não possam desfrutar da mesma alta velocidade de download atingida por outros parceiros. ■

INFORMAÇÕES

- [1] BitTorrent: <http://bitconjur.org/BitTorrent/>
- [2] Azureus, cliente BitTorrent em Java: <http://azureus.sourceforge.net/>
- [3] Especificação do Protocolo: <http://bitconjur.org/BitTorrent/protocol.html>

Diário do SysAdmin: PhpLDAPAdmin

Caso Secreto

Se você está satisfeito em usar o PhpMyAdmin para administrar o banco de dados MySQL, pode estar perdendo uma interface Web adequada para o gerenciamento do diretório LDAP.

POR CHARLY KÜHNAST

A solução (de código aberto) de que você precisa para recuperar o equilíbrio e a paz de espírito chama-se PhpLDAPAdmin, que se encontra na versão 0.9.4b [1]. O PhpLDAPAdmin está disponível como um arquivo TAR de 368 KBytes para Linux. Os requisitos são mínimos: PHP 4.1.0 ou posterior e Apache. Você precisa mover o diretório criado ao descompactar o arquivo para um caminho acessível ao Apache. Em seguida, renomeie o arquivo-exemplo de configuração para *config.php*, antes de iniciar seu editor. Inicialmente, basta adicionar o nome do seu servidor LDAP e as credenciais do usuário. Depois rode um navegador, direcione-o para a página inicial e tudo irá de vento em popa.

Fiquei extremamente impressionado com a quantidade de tarefas que o PhpLDAPAdmin consegue realizar. Existem modelos (templates) disponíveis para trabalhos repetitivos - por exemplo,

para gerenciar contas de usuários e grupos, caderno de endereços, servidores e usuários Samba e entradas de DNS. Você pode criar seus próprios modelos, caso a coleção padrão não inclua a funcionalidade de que você precisa - há até um assistente, que mostra classes pré-definidas de objetos com os quais você pode criar novos recursos (Figura 1).

Assistência da melhor qualidade

O PhpLDAPAdmin fornece o tipo de assistência de que eu preciso ao gerenciar um diretório LDAP. O PhpLDAPAdmin automaticamente seleciona o modelo correto segundo o tipo de dado que você precisa modificar. Isso significa que você tem acesso a todo o conjunto de atributos, incluindo dados binários e imagens. Gerenciar as estruturas de informação do LDAP é muito simples. Você pode mover e copiar entradas - até mesmo entre de múltiplos servidores

LDAP, se necessário. A ferramenta também permite copiar e apagar recursivamente sub-árvores completas.

Ao exportar dados do LDAP, você pode escolher entre os formatos LDIF e DSML. Diga se o sistema de destino é Unix, Mac, ou Windows e deixe o PhpLDAPAdmin cuidar da codificação correta do caractere de fim de

linha. O PhpLDAPAdmin oferece buscas simples e avançadas de dados LDAP, sendo que a última utiliza filtros práticos - que permitem aos administradores restringir as buscas a uma única seção - e exibe atributos individuais.

O PhpLDAPAdmin respeita as características do servidor de destino, não importando se o sistema operacional é Sun, Microsoft, ou Novell. Isso é útil, uma vez que nem todos os fabricantes levam as especificações tão a sério quanto deveriam. No meu caso, o PhpLDAPAdmin funcionou perfeitamente com um servidor OpenLDAP. A interface de ferramentas do usuário fala onze línguas. Além disso, e para completar os elogios desse mês, o programa suporta até mesmo UTF-8.

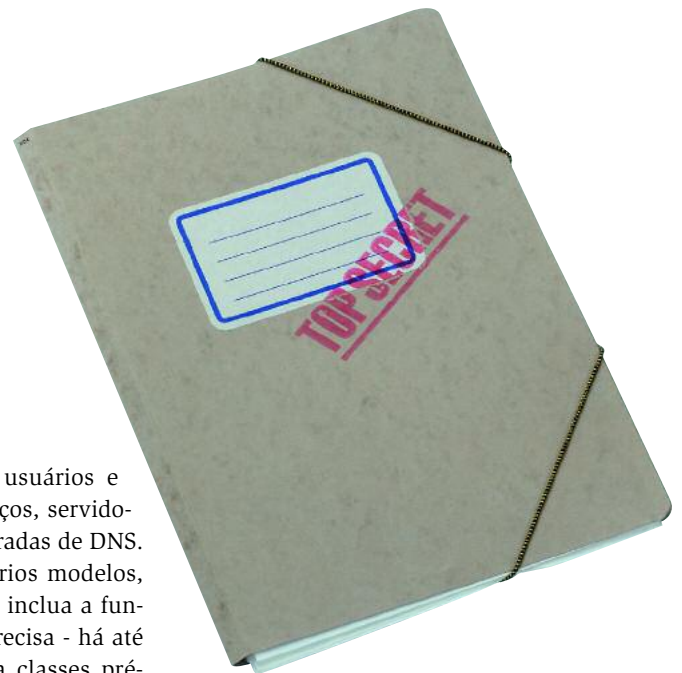


Figura 1: Clique nas classes de objetos definidas anteriormente para criar seu próprio modelo PhpLDAPAdmin.

INFORMAÇÕES

[1] PhpLDAPAdmin:
<http://phpldapadmin.sourceforge.net>

SOBRE O AUTOR

Charly Kühnast gerencia Sistemas Unix em um data-center em Moers, próximo ao famoso rio Reno, na Alemanha. Seu trabalho inclui verificação de segurança e disponibilidade dos firewalls e também é o responsável pela DMZ (demilitarized zone).



CARTAS

■ Meus parabéns

Recebam os meus parabéns pela brilhante iniciativa de trazer ao Brasil uma publicação do peso da LM. Num momento em que é crucial para a parcela informatizada da sociedade brasileira obter informações frescas e seguras sobre o SO que mais cresce em nosso país, com certeza a LM será nosso ponto de referência definitivo!

Carlos Lorenz

» Assim como o Carlos, dezenas de outros leitores escreveram para nos dar os parabéns pela nossa primeira edição. Agradecemos os elogios e o apoio, mas saibam que não pretendemos descansar sobre os louros da fama. Estamos trabalhando duro para que as futuras edições sejam ainda melhores, e para atender às várias sugestões enviadas por nossos leitores

■ CD – ROM

Não seria interessante ter uma página da revista explicando o conteúdo do CD-ROM e como usá-lo? Poderiam indicar qual o caminho para aprender Linux passo a passo para um iniciante?

Antonio Carlos Gomes - Volta Redonda - RJ

» A partir desta edição teremos, dentro da seção *Linux User*, um artigo detalhando o conteúdo do CD que acompanha nossa revista. Quanto ao aprendizado, a melhor forma é ler bastante. Leia os artigos da seção *Linux User* e guias como o *Foca Linux* (focalinux.cipsga.org.br). Se você tiver tempo, participar de listas de discussão como a *Linux-BR* também pode ser uma boa pedida. Leia também o *Guia do e-mala* (brlinux.linuxsecurity.com.br/e-mala.htm) para dicas do que você **nunca** deve fazer ao pedir ajuda em um fórum ou lista de discussão.

■ Wir sind nicht Amerikaner!

Muitos leitores nos escrevem sugerindo a publicação de artigos da Linux Magazine norte-americana (www.linuxmagazine.com), ou confusos com as diferenças na diagramação e estilo entre ela e a versão nacional. Vamos esclarecer: a Linux Magazine Brasil não tem relação com a Linux Magazine dos EUA. Somos uma filial da Linux Magazine Alemã, que também opera na Polônia, Romênia, Inglaterra, Canadá e, claro, o Brasil. Uma lista completa com os países onde atuamos e sites oficiais se encontra no expediente desta edição, na página 97.

Escreva pra gente

Dúvidas? Críticas? Sugestões? Mande e-mail para cartas@linuxmagazine.com.br. É impossível responder a todas as mensagens, mas cada uma delas é lida e analisada e, as mais interessantes, publicadas nesta seção.

Diário do SysAdmin: Serviços de DNS dinâmico

Formando Pares

Há apenas uma coisa errada com algumas conexões ADSL de baixo custo. Os provedores costumam fornecer um endereço IP temporário a cada máquina, que muda a cada conexão. Isso dificulta a provedoria de serviços (como um site pessoal, por exemplo) por parte do usuário. Serviços de DNS dinâmico podem resolver esse problema disponibilizando atribuição permanente de nome.

POR MARC ANDRÉ SELIG

Todo computador precisa de um endereço IP para se comunicar através do Protocolo da Internet (IP). Todavia, os provedores geralmente são maldosos e dão a seus clientes endereços dinâmicos. Ao iniciar uma conexão à Internet, você está utilizando um endereço arbitrário do banco de endereços do seu provedor. Isso complica a vida de administradores de servidores, já que precisam informar a seus clientes toda vez que o endereço do servidor muda.

Listagem 1

```
01 ;; QUERY SECTION:
02 ;;      www.abcxyz.com, type =
    A, class = IN
03
04 ;; ANSWER SECTION:
05 www.abcxyz.com.
    8h10m41s IN CNAME
    abcxyzsu7.abcxyz.com.
06 abcxyzsu7.abcxyz.com.
    2h39m44s IN A    136.199.85.18
```

Porém, os provedores têm boas razões para ser maldosos. O atual protocolo IPv4 usa endereços IP de quatro bytes. Teoricamente, isso lhe daria um número de endereços igual a 256 elevado à quarta potência, ou seja, uns quatro bilhões. Infelizmente, apenas uma pequena parte desses endereços está de fato disponível. Isso é devido, em parte, a características específicas do protocolo (como endereços multicast), mas principalmente a reservas de endereços (para pouquíssimas instituições) mais do que

generosas, datadas dos primórdios da Internet. O universo de endereçamento é muito restrito para uma rede de dimensões planetárias, tornando difícil encontrar blocos inteiros de endereços contíguos sem uso.

Muitos provedores recorrem a bancos de endereços com um número de entradas menor do que o número de clientes. Quando um cliente se conecta, o provedor concede a esse cliente um endereço temporário do seu banco. O IP é válido durante o tempo exato da sessão; quando ela é encerrada, o endereço imediatamente fica disponível para o próximo cliente. Em outras palavras, o provedor precisa de endereços IP suficientes para cobrir o número máximo de conexões simultâneas.

Problemas com o Servidor

Esse truque evita que faltem endereços e traz vantagens para o provedor. Infelizmente, apresenta desvantagens do ponto de vista do consumidor. Um endereço dinâmico é suficiente se você precisa apenas enviar requisições HTTP para servidores Web, mas rodar um servidor é bem diferente. Os navegadores precisariam de um método para descobrir o endereço dinâmico atual do servidor.

Essa não é uma questão que afete tipicamente grandes empresas, que têm concessão de linhas e utilizam IPs estáticos. Pequenas empresas e usuários domésticos costumam evitar a hospedagem Web em suas próprias máquinas, repassando esse trabalho para provedores especializados.



Em redes pequenas e médias, existem algumas rotinas para as quais um servidor com conexão ADSL é uma opção bastante útil. Uma empresa pode querer permitir que um consultor tenha acesso remoto a seus computadores para reduzir custos administrativos. Usuários domésticos podem usar seus celulares para se conectar a um servidor WAP a partir de seus escritórios em casa para utilizar serviços móveis de e-mail. Em uma época em que o ADSL de baixo custo e planos de acesso baseados no volume de tráfego tornaram-se comuns, é barato e faz sentido deixar as máquinas online 24 horas por dia. Mesmo assim, tanto o consultor como o usuário precisam conhecer o endereço IP atual do servidor.

Listagem 2

```
01 ;; QUERY SECTION:
02 ;;      selig.dyndns.org, type
    = A, class = IN
03
04 ;; ANSWER SECTION:
05 selig.dyndns.org.      51S IN
    A      80.128.96.209
```

DNS dinâmico para endereços dinâmicos

O Serviço de Nome de Domínio (Domain Name Service, DNS) traduz endereços IP numéricos para hostnames simbólicos. Para usar o Protocolo da Internet e acessar um computador chamado *www.abcxyz.com*, você precisa do endereço IP dele, como *136.199.85.18*. Um usuário,

ao requisitar informações do servidor, normalmente irá anotar ou memorizar o nome simbólico, *www.abcxyz.com*. O computador do usuário envia uma requisição a um servidor DNS para descobrir o endereço IP correspondente. Se o IP muda, o provedor de serviço muda a entrada no servidor DNS correspondente e, quando outro navegador procura pelo nome mais tarde, o novo endereço é devolvido.

Esse sistema é também de grande utilidade para registrar o status atual de endereços de IP dinâmicos. Você pode modificar a entrada do servidor DNS para seu próprio endereço a cada vez que ele mudar, permitindo que os clientes pesquisem o endereço mais recente ao precisarem de acesso. Todavia, há algumas coisas com as quais é preciso tomar cuidado.

Além de publicar o mapeamento de nome/endereço, os servidores DNS também atribuem prazos de validade para cada registro. Isso permite a economia de banda. Se diversos clientes requisitam o mesmo endereço dentro do período de validade determinado, o servidor DNS responderá com uma resposta em cache, sem necessidade de novo pedido a um dos servidores DNS principais.

A Listagem 1 mostra um exemplo de pesquisa manual no DNS usando *dig www.abcxyz.com*. O hostname *www.abcxyz.com* é apelido para o endereço *abcxyzsu7.abcxyz.com*. A atribuição será válida por, no mínimo, oito horas, dez minutos e 41 segundos. O endereço IP concedido a *abcxyzsu7.abcxyz.com* não irá mudar por cerca de três horas.

Listagem 3

```
01 daemon=60 # Verifica o endereço
    IP a cada 60 segundos
02 syslog=yes
03 mail=root
04 use=if, if=eth0
05 login=seligm
06 password=topsecret
07 server=members.dyndns.org,
    protocol=dyndns2
    selig.dyndns.org
```

Figura 1: Muitos roteadores possuem um cliente para o serviço DynDNS. Simplesmente utilize a interface Web para fornecer a informação necessária para permitir acessos DNS para o seu próprio servidor.

Mudanças Inesperadas

Problemas de acesso podem ocorrer no caso de mudanças inesperadas - por exemplo, se *abcxyzsu7.abcxyz.com* tem um endereço de um grupo dinâmico e ele muda antes que as 2 horas, 39 minutos e 44 segundos tenham expirado. Cliente continuarão usando a resposta em cache, assim como qualquer outra máquina que acessar o mesmo servidor DNS. Já os servidores sem entrada no cache para *www.abcxyz.com* criariam imediatamente uma nova entrada.

A resposta para esse dilema é conceder uma validade extremamente curta para as entradas de DNS dinâmicas. Infelizmente, isso pode causar um aumento drástico no tráfego para o servidor DNS. Olhando pelo lado bom, isso significa um menor potencial de erros em seguida à mudança do endereço IP. A Listagem 2 mostra um exemplo típico de uma entrada dinâmica. A saída de *dig selig.dyndns.org* nos diz que a atribuição de endereço é válida apenas por 51 segundos. Na realidade, o servidor de nomes tende a apagar aquela entrada do cache ainda antes disso.

Um servidor DNS que publica um endereço dinâmico não pode ter um endereço dinâmico para si mesmo. Isso seria como a história do ovo e da galinha. Felizmente, existe um bom número de grandes provedores de serviço DNS que mapeiam endereços de IP dinâmicos a nomes estáticos. Eu uso o DynDNS [1] e NO-IP [2], mas existem muitos outros serviços similares.

O próximo cuidado que você deve tomar é fazer com que a sua máquina atualize suas próprias informações DNS.

Os arquivos de configuração que os administradores normalmente fornecem aos seus servidores DNS são um tanto complexos. Seria por demais complicado atualizar os arquivos manualmente toda vez que seu IP mudasse. Um software especial para os clientes, que transmita automaticamente o novo IP para o servidor DNS sempre que a máquina inicia uma conexão à Internet, faz mais sentido.

Atualização Automática

Muitos roteadores projetados para o uso em pequenas empresas ou escritórios domésticos têm um cliente para serviços DNS dinâmicos. (veja a Figura 1). Um sistema Linux pode fazer o mesmo serviço com igual facilidade. Não faz diferença se você acessa a Internet através de um roteador ou utiliza os recursos de roteamento do Linux com um modem ISDN, ADSL ou outra tecnologia. A escolha do software cliente depende do provedor DNS.

A ferramenta *ddclient* [3] é um programa popular. Muitas distribuições Linux já incluem o cliente, cuja configuração é feita no arquivo */etc/ddclient.conf* (veja Listagem 3). O programa roda como um daemon e vigia a interface de rede, procurando por mudanças de endereço IP em intervalos regulares (linha 4); 60 segundos em nosso exemplo, como mostrado na linha 1.

Toda vez que o endereço muda, o *ddclient* atualiza as entradas do servidor DNS, usando para isso o protocolo e o hostname indicados no arquivo de configuração (linha 7). Isso faz com que o servidor DNS responda com o endereço IP atual da máquina sempre que recebe uma requisição para *selig.dyndns.org*. Isso, por sua vez, permite que a máquina continue a oferecer serviços. ■

INFORMAÇÕES

- [1] Um dos serviços mais antigos de DNS dinâmico: <http://www.dyndns.org>
- [2] Outro serviço de DNS dinâmico: <http://www.no-ip.com>
- [3] Cliente Perl para o DynDNS: <http://www.dyndns.org/services/custom/clients.html##3>

Configurando um repositório APT para o Debian

Espelho, espelho meu...

Graças à ferramenta APT, a atualização de um sistema Debian pode ser feita de forma rápida e fácil. No entanto, todos os pacotes são baixados de servidores externos, previamente indicados em um arquivo de configuração. O tráfego gerado por essa operação pode tornar-se uma dor de cabeça para os administradores, dependendo da quantidade de sistemas Debian na rede. Um “espelho” replicando localmente o conteúdo destes servidores pode ajudar a resolver o problema.

POR BRUNO BELLANTUONO

A necessidade de manter diversas máquinas Debian sempre atualizadas, sem congestionar o link da empresa, motivou-me a procurar uma alternativa. Criar um repositório interno com os pacotes de software, atualizado diariamente, torna o processo de instalação extremamente rápido e garante a presença de todas as correções e atualizações de software.

A própria página oficial do Debian menciona dois meios de criar um “espelho” (espelho) de um servidor APT: uma é através de um script que usa o utilitário `rsync`, a outra através do pacote `debmirror`, escrito por Joey Hess e Joerg Wendland, que será mostrado neste artigo. O `debmirror` é mais fácil de usar e pode ser encontrado em qualquer repositório de pacotes do Debian.

Espaço em disco

É muito importante escolher bem o que baixar dos servidores principais. O repositório oficial possui binários para as mais diversas arquiteturas, desde a PowerPC (comumente usada em Macs e máquinas da IBM) até processadores de 64 bits. Isto determina o espaço em disco necessário para armazenar os arquivos. A Tabela 1 mostra dados oficiais do projeto Debian sobre os tamanhos dos “espelhos” de acordo com a arquitetura e pacotes selecionados.

Tipos de repositórios

Na Internet [2] é possível obter uma lista com todos os tipos de repositórios Debian. Existem basicamente deles:

- **Push-Primary:**

Esses sites espelham-se diretamente em um repositório mestre (que não é publicamente acessível) utilizando a técnica de “*push mirroring*”, método de espelhamento que não será abordado neste artigo.

- **Push-Secondary:**

São sites que se espelham em servidores *Push-Primary*, também usando o método *push mirroring*.

- **leaf:** Método utilizado pela maioria dos servidores.

O mais adequado é espelhar um servidor *Push-Primary*. Existe um pacote chamado *netselect*, que ajuda a descobrir o mirror mais rápido. Para instalá-lo, basta executar o comando `apt-get install netselect`. A utilização desse programa é bastante simples, mas será abordada, pois foge do escopo deste artigo.

O processo de instalação (testado em uma máquina com o Debian 3.0 “Woody”) é dividido em seis partes: Instalação do `debmirror`, o processo de espelhamento em si, criação de scripts, configuração do cron, instalação e configuração do servidor web Apache e configuração do arquivo `/etc/apt/sources.list`.



Kerry Fearby - www.sxc.hu

Instalando o pacote debmirror

Apesar de estar disponível apenas para a distribuição instável (*Debian Unstable*, codinome *Sid*), o pacote `debmirror` não possui muitas dependências e pode ser

Tabela 1 - Tamanho de um repositório Debian

Todo o repositório:

Aproximadamente 100 GB (sim, 100 Gigabytes).

Uma arquitetura oficialmente suportada:

(pacotes *stable*, *testing* ou *unstable*) De 5,5 GB a 8 GB para cada versão.

Pacotes independentes de arquitetura:

Aproximadamente 8 GB.

Quadro 1 - Exemplo de script para espelhament

```
#!/bin/sh
DATA=`date -I`
DISTS="--dist=woody,sarge,sid"
DISTS_NONUS="--dist=woody/non-US,sarge/non-US,sid/non-US"
OPTIONS1="--nosource -p -v"
OPTIONS2="--nosource -p -v --ignore=debian-non-US"
REMOTE_DIR="--root=/debian"
REMOTE_DIR_NONUS="--root=/debian-non-US"
LOG_DIR="/home/bruno/log/$DATA.main"
LOG_DIR_NONUS="/home/bruno/log/$DATA.non-us"
HOST="-h ftp.de.debian.org"
HOST_NONUS="-h ftp.de.debian.org"
#Espelha o repositório completo, ignorando o diretório debian-non-US
debmirror $HOST $OPTIONS2 $REMOTE_DIR $DISTS /home/debian >> $LOG_DIR
#Baixa somente os arquivos non-US no diretório /home/debian/debian-non-US
debmirror $HOST_NONUS $OPTIONS1 $REMOTE_DIR_NONUS $DISTS_NONUS ↵
/home/debian/debian-non-US >> $LOG_DIR_NONUS
```

instalado sem maiores problemas em sistemas baseados no Debian Woody ou Sarge. Basta baixar o pacote (veja o site na referência [3]) e instalá-lo com o comando abaixo:

```
dpkg -i debmirror_20040427_all.deb
```

O processo de espelhamento

O download dos pacotes foi feito pelo usuário *debian* e os arquivos foram salvos no diretório */home/debian*. Se você ainda não criou esse usuário, execute os comandos *adduser debian* e *passwd debian* para criá-lo e atribuir uma senha.

Nosso repositório irá conter os pacotes das versões *stable*, *testing* e *unstable* para a arquitetura x86 (i386), além do código-fonte. O exemplo usa o servidor *ftp.de.debian.org*, mas é recomendado usar o comando *netselect* para descobrir qual o servidor mais rápido em seu caso. Execute o *debmirror* da seguinte forma:

```
debmirror -h ftp.de.debian.org
--dist=woody,sarge,sid -p /home/↵
debian
```

Uma breve explicação sobre os parâmetros utilizados no comando acima:

- *-h*: Especifica o endereço do servidor de onde serão baixados os arquivos.
- *--dist*: Especifica quais distribuições serão baixadas. A distribuição padrão é a Sid. Os nomes devem ser separados por vírgulas.
- *-p*: Mostra o progresso do download.

O repositório também irá conter os pacotes “non-us”, que devido a aspectos legais não podem ser utilizados dentro do território norte-americano. Antes de baixá-los, devemos criar o diretório *debian-non-us* dentro de */home/debian*:

```
mkdir /home/debian/debian-non-US
```

Execute o comando abaixo para baixar os arquivos “non-us”:

```
debmirror -p -h=ftp2.de.debian.org \
--root=/debian-non-US --dist=woody/\
non-US,sarge/non-US,sid/non-US \
/home/debian/debian-non-us
```

Note que, em relação ao comando anterior, as diferenças restringem-se apenas ao parâmetro `--root`, que especifica o diretório no servidor remoto onde estão localizados os arquivos e os valores passados para o parâmetro `--dist`.

Criando um script

Os dois comandos executados anteriormente podem ser colocados em um Shell Script que será executado diariamente usando o *cron*. Veja no Quadro 1 um exemplo de como pode ficar o script.

É importante ressaltar que na primeira ocorrência do comando *debmirror* é necessário informar ao programa que o conteúdo do diretório *debian-non-US* deve ser ignorado (`--ignore=debian-non-US`); caso contrário, durante a atualização do repositório, o diretório */home/debian/debian-non-US* será sobrescrito. Salve o script com um nome qualquer (em nosso exemplo, *script.sh*) no diretório de sua preferência.

Configuração do CRON

Os espelhos geralmente são atualizados todos os dias às 00:00 UTC; no entanto, isso não é uma regra geral. É aconselhável atualizar seu espelho todos os dias no horário que lhe for mais conveniente.

Neste exemplo vamos atualizar nosso mirror todos os dias às 22:45. Para isso precisamos configurar o *crontab* do usuário *root* para que execute o script acima diariamente, no horário especificado. Como *root*, execute o comando *crontab -e*, e insira a linha abaixo ao final do arquivo:

```
45 22 * * * /root/script.sh
```

Salve o arquivo e saia do editor. Para nos certificarmos de que o processo está mesmo agendado, execute o comando *crontab -l*. A resposta deve conter, entre outras, a linha que acabamos de adicionar ao arquivo.

Quadro 2 - Arquivo sources.list

```
deb http://192.168.100.100/debian-non-US/ woody/non-US main contrib \
non-free
deb http:// 192.168.100.100/debian/ woody main non-free contrib
deb-src http:// 192.168.100.100/debian/ woody main non-free contrib
deb-src http://200.246.93.195/debian-non-US/ woody/non-US main \
contrib non-free
```

Instalação do servidor web

Neste ponto você já deve estar com todos os arquivos do espelho no seu disco. Portanto, chegou a hora de disponibilizar o repositório para os usuários da rede interna. Podemos fazer isso via Web (HTTP) ou FTP; aqui vamos usar o método via Web. Para isso, você precisa do servidor Web Apache. Caso ele ainda não esteja instalado, execute:

```
apt-get install apache
```

Após a instalação é necessário editar o arquivo de configuração do Apache. Abra o arquivo */etc/apache/httpd.conf* e configure os seguintes parâmetros:

```
DocumentRoot /home/debian
<Directory /home/debian/>
```

Além disso adicione as seguintes linhas ao arquivo de configuração do Apache:

```
Options +Indexes \
+SymlinksIfOwnerMatch
IndexOptions NameWidth=* \
+SuppressDescription
DirectoryIndex .
```

Isso ativa os índices de diretório e garante que os *symlinks* (links simbólicos) irão funcionar. Os nomes de arquivo nos índices de diretório não ficarão truncados e as descrições (em geral não existentes) não serão mostradas.

O arquivo de configuração

Agora que o seu espelho já está devidamente configurado, precisamos apontar o arquivo */etc/apt/sources.list* das máquinas locais para o nosso novo servidor. Considerando que o servidor tenha o endereço IP 192.168.100.100 e que o computador cliente use a distribuição estável (*stable* ou *Woody*), o arquivo */etc/apt/sources.list* deve ficar como mostrado no Quadro 2.

As duas primeiras linhas apontam para os pacotes binários em nosso servidor, e as duas últimas para o código-fonte desses pacotes.

A presença de um espelho dentro de sua rede local facilita o processo de atualização e instalação de novas máquinas. Caso você tenha largura de banda sobrando e deseje disponibilizar seu espelho para toda a comunidade Debian, basta preencher o formulário [4] no site do Debian.

Este artigo foi baseado em informações obtidas em vários sites da Internet, principalmente na página oficial do Debian. Para saber mais sobre o assunto, consulte os vários links no quadro *Informações* logo abaixo. ■

INFORMAÇÕES

- [1] Site oficial do Debian:
<http://www.debian.org>
- [2] Lista oficial de Mirrors:
<http://www.nl.debian.org/mirror/list-full>
- [3] Pacote debmirror:
http://ftp.us.debian.org/debian/pool/main/d/debmirror/debmirror_20040427_all.deb
- [4] Divulgue seu mirror:
<http://www.nl.debian.org/mirror/submit>
- [5] Página oficial do Debian sobre “mirroring”:
<http://www.nl.debian.org/mirror/ftpmirror>
- [6] Um How-To não oficial:
<http://www.mail-archive.com/vox-tech@lists.lugod.org/msg04544.html>

SOBRE O AUTOR

Bruno Bellantuono trabalha na Cyclades, onde é o responsável pelo desenvolvimento dos manuais dos produtos da empresa. É um dos mantenedores do site Linux Temple, em www.linuxtemple.com, com notícias e artigos sobre o Linux.



A próxima geração

Feitiço da Lua

Como linguagem de script, Lua é bastante única. Tem uma grande aceitação de dois nichos do mercado: usuários Linux e programadores de jogos para computadores. O que há de atraente nessa linguagem? O que é tão especial? Steven Goodwin investiga. **POR STEVEN GOODWIN**

De um ponto de vista lingüístico, a Lua pode ser considerada bem comum! Tem todas as características necessárias para torná-la uma linguagem utilizável (variáveis, funções e controle de fluxo), mas carece de alguns pontos importantes na linha das expressões regulares do Perl ou da velocidade de execução do C. Todavia, sua força não reside nas partes individuais da linguagem, mas em como funciona no geral e em como se conecta ao mundo externo.

Neste artigo, olharemos para Lua como uma ferramenta de personalização e configuração, mostrando como o usuário final pode personalizar certos softwares utilizando Lua (nós vamos usar o *elinks* como um exemplo) e que passos o programador deve dar para implementar essa funcionalidade em seus próprios aplicativos. Mas antes, teremos a costureira aula de história...

A lua como um todo

Lua começou em 1993 como uma linguagem para acadêmicos brasileiros da Tecgraf, na Pontifícia Universidade Católica do Rio de Janeiro (PUC-RJ). Sua proposta era fornecer um método simples de estender aplicativos através de uma linguagem procedural básica, tipos de dados tradicionais e uma máquina virtual. Todas essas características continuaram a ser parte fundamental de *Lua* até os dias de hoje, com a versão 5.0.2 (que contém correções para pequenos erros) no dia 17 de março de 2004. Sua assimilação foi uniforme (alguns diriam vagarosa), assim como seu ciclo de atualização; apenas 12 versões públicas em 11 anos de história. Em compensação, há muito pouca chance de quebra repentina do código, e todas as versões, até agora, têm sido incrivelmente estáveis.



A estrutura básica de Lua é a de um interpretador (o programa, *lua*) que gera e executa byte code, um conjunto de bibliotecas básicas opcionais (para entrada e saída, funções matemáticas e assim por diante, tudo escrito em C padrão) e um compilador (*luac*) para gerar byte code offline. Devido à natureza altamente padronizada da base de código de Lua, ela funciona em quase todas as plataformas que comportam um compilador C. Uma vez que é modular, pode encaixar-se em muitos dispositivos embutidos sem ocupar um espaço grande e desnecessário. A versão Linux, por exemplo, roda com apenas pouco mais de 100K; o conjunto de bibliotecas padrão incluído ocupa mais 72K. Muitos

projetos embutidos tiraram vantagem de seu pequeno tamanho (e sua habilidade de compilar em qualquer lugar), como foi o caso de muitas empresas de jogos, como a Criterion Studios, a LucasArts (Grim Fandango), e a BioWare (com MDK2 e Baldur's Gate). Para uma lista mais longa dos usos, sugiro uma visita à referência [1].

A lua cheia da primavera

Como linguagem, ela tem todas as características esperadas em uma linguagem funcional de script, incluindo o obrigatório manual online, disponível em [2]. Em primeiro lugar, nós temos os tipos de dados. Talvez não sejam abundantes (veja Tabela 1: Tipos em Lua), mas são

Tabela 1: Tipos em Lua

Tipo	Identificador	Nome para lua_istype
Nil	LUA_TNIL	nil
Number	LUA_TNUMBER	number
Bool	LUA_TBOOLEAN	boolean
String	LUA_TSTRING	string
Table	LUA_TTABLE	table
Function	LUA_TFUNCTION	cfuction
Userdata	LUA_TUSERDATA	userdata

Notas

- O identificador pode ser usado para traduzir a constante em uma cadeia de caracteres dentro do C, usando `lua_typename(lua_State *L, int type)`
- `lua_isnumber` aceita os números (123) e cadeias de caracteres numéricos ("123")
- `lua_toboolean` retorna o para `false` e `nil`, e `1` para os demais

capazes de satisfazer às necessidades normais dos programadores. As variáveis em si seguem os moldes de uma linguagem flexível e podem comportar qualquer tipo, em qualquer momento. A tentativa de usar variáveis indefinidas faz com que elas contenham o tipo `nil`, que pode fazer com que certas operações (por exemplo, concatenação de cadeias de caracteres) falhem. O conceito de um "número" em Lua é equivalente ao tipo `double` da linguagem C. Todavia, viciados

série tradicional de instruções de controle de fluxo, nos seguintes padrões:

```
do block end
while exp do block end
repeat block until exp
if exp then block end
if exp then block1 else block2 end
if exp then block1 elseif exp then
block2 end
for var=start,end[,step] do block end
for var in iterator do block end
```

em desempenho que estejam usando a versão 5 podem modificá-lo para `float` (ou mesmo `int`) e recompilar `Lua`. Adicione:

```
#define LUA_NUMBER float
```

antes de incluir `lua.h`. Quem ainda utiliza a versão 4 terá que modificar mais um pouco o código.

Há também uma

Nada diferente aqui, mas programadores de C devem estar cientes de que o valor de 'end' é também calculado no loop `for`. Note que todas as instruções utilizam a palavra `end` como um finalizador de bloco, ao invés das chaves, mais tradicionais. Essa simplicidade é uma tentativa bastante óbvia de empurrar os não programadores para o mundo do script. A preferência por palavras ao invés de símbolos é também aparente quando você percebe que os operadores `!`, `&&` e `||` foram substituídos pelas seus equivalentes alfabéticos, `not`, `and` e `or`.

A sintaxe também remove algumas das restrições encontradas em outras linguagens. Por exemplo, dois (ou mais) parâmetros podem ser retornados de funções, sem quaisquer problemas ou magia negra.

```
function onetwo()
return 1,2
end
one,two = onetwo()
```

Finalmente, em nosso rápido tour, Lua contém funções locais e anônimas. Uma função local é aquela que apenas pode ser chamada de dentro da função em que ela declaradamente está. Isso não é comum para os programadores de C, mas é bastante familiar para todos os outros. Já as funções anônimas permitem embutir uma função inteira no lugar em que nós normalmente colocaríamos uma chamada de retorno. Isso evita a necessidade de funções adicionais e arbitrárias dentro do código.

Para informações mais detalhadas sobre a sintaxe da linguagem, é um boa idéia ler a versão original disponível no website de Lua em [2]. O fórum de discussão em [3] está disponível para tópicos mais avançados. Porém, os fóruns podem receber um pouco de spam, de tempos em tempos.

Listagem 1: hello.lua

```
01 // Insira aqui um comentário
    irônico sobre a originalidade
    do programador!
02 function hello()
03 write("Hello, Lua!")
04 end
05 hello()
```

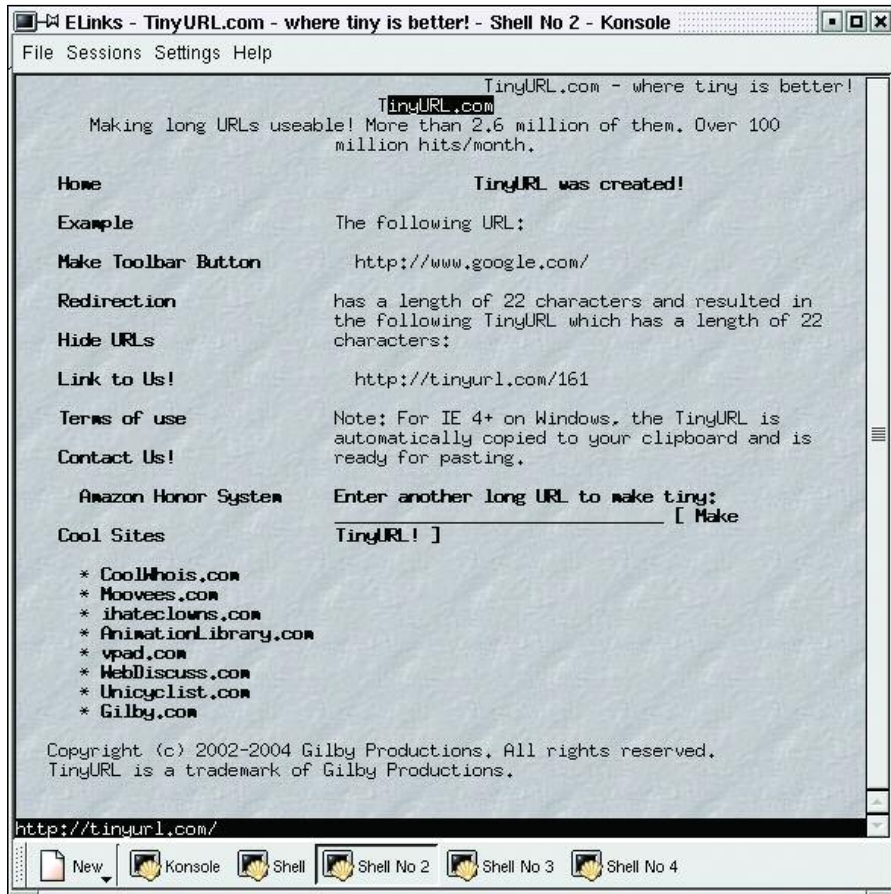


Figura 1: O resultado da sua função gancho.

Todavia, não são as características de Lua como linguagem o que a faz ser aprovada pelos desenvolvedores. É principalmente seu uso como ferramenta de configuração, que é considerado sua maior aplicação. É tão fácil para um desenvolvedor de aplicativos adicionar o suporte a scripts em Lua que seria possível perguntar por que a linguagem não é mais comum. Pode ser utilizada para criar plug-ins para softwares ou como um interpretador de arquivos de configuração. Não precisa ser uma configuração estática, como a maioria das outras aplicações. Ao utilizar uma linguagem de programação para gerar a configuração, você pode criar algo muito mais poderoso e flexível. Pode criar algo dinâmico!

Configurações dinâmicas têm sido um tanto raras. Apenas os aplicativos mais complexos, como o Apache, as possuem. Mesmo assim, diretivas como *IfModule* são reduzidas ao mínimo e têm um alcance limitado. Uma configuração verdadeiramente dinâmica pode facilitar o processo de instalação, preparando-se quando o programa é executado e adaptando-se de acordo com a estrutura do diretório, número de usuários, atual largura da banda, carga do processador e assim por diante.

Lua também disponibiliza um método fácil de adicionar ganchos (“hooks”) no software para personalização. Veremos esse atributo a seguir, adicionando alguns ganchos simples ao navegador de texto *elinks*.

Caminhando à luz da lua

O gancho é um método através do qual um programa (nesse caso, o *elinks*) chama uma função especial toda vez que está para fazer alguma coisa importante. Esse ‘coisa importante’ pode ser ir para uma URL ou baixar uma página HTML do servidor. Em situações normais, sem gancho, essa função especial não fará nada! Nada mesmo! Ponto! Ela simplesmente devolverá o controle para o programa principal e o deixará ir para a URL para a qual estava indo anteriormente.

Entretanto, quando um gancho é colocado nessa função especial, o controle é passado do programa principal para a função gancho. Nessa altura, esta função tem o controle das informações e pode reescrever a URL, por exemplo. Como esses ganchos foram programados a par-

Tabela 2: Ganchos

Função Gancho	Chamada quando...	Deve retornar...	Notas
<code>goto_url_hook(url, current_url)</code>	Uma URL é digitada no diálogo “Go to URL”	Uma nova URL, ou zero para cancelar	
<code>follow_url_hook(url)</code>	Uma URL foi selecionada	Uma nova URL, ou zero para cancelar	
<code>pre_format_html_hook(url, html)</code>	Um documento foi baixado	Seqüência modificada, ou zero se nenhuma mudança foi feita	Pode Remover anúncios/lixo das webpages mal concebidas
<code>lua_console_hook(string)</code>	Algo é digitado no console de Lua (<i>type</i> , <i>in elinks</i>)	Um comando (<i>run, eval, goto-url, or nil</i>) seguido de um argumento apropriado (o programa para executar, o código Lua para validar, uma URL para ir, ou nulo, respectivamente)	
<code>quit_hook()</code>	<i>elinks</i> está para terminar	Nada	Para organizar os recursos

tir de um programa Lua - o *nosso* - podemos reescrevê-los de acordo com nossas preferências pessoais.

Por exemplo, eu posso querer visitar um de meus próprios websites, *www.BlueDust.com*, digitando *bd*. Eu posso fazê-lo criando um gancho simples na rotina ‘Jump to URL’ com:

```
if (url == "bd") then
return "www.bluedust.com"
end
```

Vejam só, configuração instantânea!

O programa usado para demonstração neste artigo, o *elinks*, tem uma variedade de ganchos diferentes, que podem ser vistos na Tabela 2: Ganchos e Frestas. Nós podemos utilizar um ou alguns deles para personalizar o aplicativo.

Não é só o *elinks* que chama nosso script, é possível para nosso script chamar o *elinks* através das chamadas de retorno. Isso nos permite resgatar informações, como a página de título, que não são repassadas como parâmetro. Isso é especialmente útil com teclas de atalho, como: quais parâmetros devem ser transferidos para elas? URLs? Listas de Favoritos? É aí que entram as funções de chamadas de retorno.

Essas chamadas de retorno são funções específicas que o *elinks* decidiu

que nós (enquanto script) devemos utilizar. Ele nos permite chamá-las como se fizessem parte do próprio script em Lua. Incluem funções como *current_url* e *current_title*. Veja a lista na Tabela 3. Devemos agora empregar ambas as idéias para criar um gancho simples para o *elinks* que crie uma pequena URL para a página atual.

A dança da lua

Nossa primeira tarefa é nos certificarmos de que o *elinks* foi de fato compilado com o suporte para scripts em Lua. Você pode conferir isso abrindo a caixa *About*: Pressionando *Alt + H*, seguido da letra *A*. Entre todo o texto, você deve encontrar as seguintes palavras:

Scripting (Lua)

Quadro 1: enable systems functions

openfile, closefile, readfrom, writeto, appendto, pipe_read, remove, rename, flush, seek, tmpname, read, write execute, exit, clock, date, getenv, setlocale.

Listagem 2: goto_url_hook

```
01 function goto_url_hook (url, current_url)
02 if url == "tiny" then
03   return
   "http://tinyurl.com/create.php?url="..current_url
04 end
05 return url
06 end
```

Isso já é normalmente incluído na maioria das distribuições (ou baixado em [4]). Casos o suporte não exista, corrija o problema recompilando o *elinks* com:

```

$./configure --with-lua
$ make
# make install
    
```

Isso também irá criar um arquivo de exemplo chamado *hooks.lua* no diretório *elinks/contrib/lua*. Depois, leia tudo com atenção e descubra algumas das outras possibilidades de script em Lua.

Em seguida, precisamos criar um script que será executado quando o *elinks* iniciar. Ele é colocado em *~/elinks/hooks.lua* e é executado integralmente durante a inicialização. Então, desde que nosso código esteja dentro de *functions*, nada deve aparecer na tela.

A primeira de nossas funções irá incluir o código para *goto_url_hook*. Como foi anteriormente mencionado, ele é chamado toda vez que o usuário digita 'g' para mudar a webpage. É portanto bem simples escrever a Listagem 2.

É realmente fácil como aparenta! Recarregue o *elinks* e viste seu último website não favorito. Depois pressione 'g', seguido pela palavra-chave *tiny*, e retorne. Se você é como eu e escolhe o Google como seu site de teste, será levado para uma webpage em *tinyurl.com*, como mostrado na Figura 1.

Podemos ver a nova URL designada como *http://tinyurl.com/161*, que pode depois ser copiada, colada, enviada por e-mail e mal-utilizada de maneira geral. Se nós soubéssemos como adicionar atalhos no *elinks*, poderíamos poupar quatro teclas de atalho. Aqueles que leram anteriormente a Tabela 3, conhecerão a função chamada *bind_key*. Com ela, podemos adicionar o código mostrado na Listagem 3.

O exemplo demonstra a utilidade de funções anônimas e a facilidade com que

dois valores podem ser retornados de uma função. No caso, o comando *goto_url* e o parâmetro URL.

Para melhorar o acabamento, vamos remover a informação duplicada da URL, escrevendo o arquivo *hooks.lua* como na Listagem 4.

Como você pode ver, Lua facilita ao usuário final adicionar funcionalidades em uma parte do software. Você não precisa de nada além dos métodos fornecidos aqui e um pouco de imaginação para adicionar um monte de outras funcionalidades. De um modo geral, dar mais flexibilidade ao usuário final significa complexidade para o programador. Com Lua, isso não é o que acontece! Vamos ver o porquê...

Foguete Lunar

Em qualquer sistema extensível como esse, existem três componentes básicos. A inicialização (e desligamento), a comunicação de entrada (em que o script comunica-se com o nosso programa em C), e a comunicação de saída (em que o C comunica-se com o script). Todas as três áreas são muito simples e podem utilizar os modelos básicos mostrados aqui. Essa simplicidade manteve o interpretador Lua pequeno e encorajou os programadores a utilizarem-no em várias tarefas, como para configuração e automação de tarefas.

Vamos começar pelo início. Veja o exemplo a seguir:

Listagem 4: Arquivo *hooks.lua*

```

01 function get_tiny(url)
02   return "http://tinyurl.com/create.php?url="..url
03 end
04 function goto_url_hook (url, current_url)
05   if url == "tiny" then
06     return get_tiny(current_url)
07   end
08   return url
09 end
10 bind_key ("main", "Ctrl-T", function () return
    "goto_url", get_tiny(current_url()) end )
    
```

```

#include <lua.h>
int main(int argc, char *argv[])
{
    lua_State *pLua;
    pLua = lua_open();
    printf("Hello, World!");
    lua_close(pLua);
    return 0;
}
    
```

A primeira coisa a ressaltar é que estou executando o Lua 5. Mesmo que essa seja uma preferência pessoal, pode causar pequenos problemas de compatibilidade, já que *lua_open* precisa de um parâmetro do tamanho da pilha na versão 4. Aqueles que executam a versão 4 (que inclui a versão estável do Debian) também irão notar que o arquivo de cabeçalho deve ser modificado para *lua40/lua.h*. Essas são duas das poucas mudanças incompatíveis com as versões

Tabela 3: Funções

Nome da Função	Propósito
<i>enable_systems_functions()</i>	Permite que certas funções (por exemplo, abrir arquivo) sejam usadas. Veja Caixa 1.
<i>current_url()</i>	Recupera a URL da página atual do <i>elinks</i>
<i>current_link()</i>	Recupera o link que está selecionado no momento (ou <i>nil</i> , se nenhum)
<i>current_title()</i>	Recupera o título da página
<i>current_document()</i>	Recupera a página HTML, como uma cadeia de caracteres
<i>current_document_formatted([width])</i>	Recupera a página HTML, formatada com o tamanho opcional
<i>pipe_read(command)</i>	Executa o comando dado e lê dados
<i>execute(command)</i>	Executa o comando dado em shell (using <i>sh -c</i>)
<i>bind_key(keymap, keystroke, function)</i>	Executa a função toda a vez que uma tecla de atalho é criada. Deve retornar um comando e um parâmetro como <i>lua_console_hook</i>

Listagem 3: Função *bind_key*

```

01 bind_key ("main", "Ctrl-T",
02   function ()
03     return "goto_url",
04     "http://tinyurl.com/create.php?url="..current_url()
05   end
06 )
    
```


Listagem 5: rotina *factorial*

```
01 bind_key ("main", "Ctrl-T",
02  function ()
03      return "goto_url","http://tinyurl.com/create.php?url="..current_url()
05  end
06  )
```

anteriores feitas na versão 5. Ao compilar, devemos *linkar* nosso código C com a biblioteca de Lua da seguinte forma:

```
$ gcc -llua mycode.c
```

Se você está utilizando apenas Lua (isto é, uma versão standalone, não embutida em uma aplicação), precisará de suas próprias rotinas de entrada e saída. Isso não está incluído como padrão, uma vez que a maioria dos aplicativos provê suas próprias rotinas de E/S e, dessa forma, eliminam uma carga desnecessária. Para permitir ao código de Lua acessar suas bibliotecas padrão, inclua o seguinte código em C:

```
lua_baselibopen(pLua);
lua_iolibopen(pLua);
lua_strlibopen(pLua);
lua_mathlibopen(pLua);
```

A escolha das bibliotecas necessárias cabe, obviamente, a você. Todavia, incluir qualquer uma delas necessita vinculação com a biblioteca *lua.lib*. Utilizar a biblioteca de funções matemáticas de Lua requer a biblioteca de funções matemáticas do C (que não está incluída como padrão), e a linha de compilação vai se parecer com a seguinte:

```
$ gcc -llua -llualib -lm ?
mycode.c
```

O *pLua* contém o estado de todo o sistema Lua. Uma vez que Lua é reentrante, podemos chamar *lua_open* quantas vezes nós quisermos, e nenhum estado irá entrar em conflito com o outro, o que nos permite usar Lua como parte de um sistema integrado. Toda vez que interagirmos com Lua, devemos utilizar esse ponteiro, que é convencionalmente chamado *L*, embora no texto eu esteja utilizando *pLua* para maior clareza.

Tendo acesso ao estado de Lua, podemos alimentá-lo com código para ser processado pelo interpretador interno:

```
lua_dostring(pLua, "number=12345");
// other code
lua_dostring(pLua, "print(number)");
```

Enquanto mantivermos o controle do *pLua*, quaisquer variáveis definidas permanecerão no estado de Lua. Toda vez que chamarmos uma função (como *print*), Lua irá validá-la com as funções declaradas e retornar os resultados para o estado mantido em *pLua*.

Provavelmente você pode imaginar como seria fácil construir seu próprio interpretador e depurador com apenas essa simples função, e você está certo! Porém, tal função já foi escrita para nós. É chamada *dofile*. Ela executa o código dentro do arquivo como se ele estivesse sendo executado a partir da linha de comando: isto é, executando apenas aquelas instruções que são globais. Todavia, diferentemente da linha de comando, assim que o arquivo for executado, o estado permanece em *pLua*. Essas variáveis, junto com quaisquer funções declaradas, podem ser acessadas pelo código em C, por outro arquivo Lua carregado com o *lua_dofile* ou pelo código validado com *lua_dostring*.

```
int result;
result = lua_dofile?
(pLua, "config.lua");
```

Nesse caso, *result* retorna o efeito do último segmento, que normalmente depende da execução bem-sucedida ou não o script.

Luar sobre Parador

Vamos agora escrever algo útil em nosso script *config.lua*, que chama uma função em nosso programa em C. Isso teria o mesmo efeito que a função *current_url* no *elinks*, por exemplo.

Para fazer isso, precisamos registrar uma de nossas funções em C com Lua. Isso une as duas linguagens. Damos o nome que Lua usaria, seguido pelo nome atual de nossa função em C.

Listagem 6: Chamando Lua

```
01 // O nome da função é um símbolo global: nós devemos usá-lo em vez
    de uma cadeia de caracteres contendo o nome da função
02 lua_getglobal(pLua, "swap_greeting");
03 // Nosso primeiro parâmetro
04 lua_pushstring(pLua, "Steev");
05 // Nosso segundo parâmetro
06 lua_pushstring(pLua, "Hello");
07 // A chamada em si
08 lua_call(pLua, 2/*number of input args*/, 2/*number of result args*/);
09 // Recuperar resultados dentro das variáveis de Lua
10 lua_setglobal(pLua, "result2");
11 lua_setglobal(pLua, "result1");
```

```
lua_register(pLua, "factorial", 2
c_factorial);
```

A partir daqui, podemos deixar que Lua tome conta. Ela organizará todas as chamadas de funções, passagem de parâmetros e o retorno dos resultados. Tudo o que nós precisamos fazer é recuperar os parâmetros na ordem certa e retornar os resultados corretos.

Já que Lua comporta múltiplos parâmetros de retorno (e de qualquer tipo), nós não podemos utilizar qualquer protótipo típico do C para administrá-la. No lugar disso, todos os parâmetros são enviados (e extraídos) de uma pilha. Nós podemos então consultá-la para saber quantos itens estão nela e de que tipo são. As pilhas, como variáveis, podem conter qualquer tipo de dado suportado. Cabe a nós, como programadores em C, requisitar o tipo apropriado ao remover dados da pilha. Isso garante que cada função do C tem a mesma assinatura, ou protótipo, quando usada com Lua.

Se nossa rotina *factorial* pode pegar um número arbitrário de números inteiros e retornar o fatorial de cada um,

podemos recuperar o número de argumentos e computar o resultado para cada um - desde que seja numérico. Nosso código pareceria um pouco com o da Listagem 5.

Observe que os índices da pilha são contados a partir do 1 (não 0) e que precisamos retornar o número de parâmetros que são enviados para a pilha de retorno. Fora isso, não há mágica alguma em escrever suas próprias funções. Os tipos podem variar, de forma que *lua_isnumber* deve tornar-se *lua_isbool* (veja Tabela 1: Tipos de Lua), mas os princípios são exatamente os mesmos.

Nuvens cruzando a lua

Chamar uma função de Lua a partir do C não é mais difícil, se você souber o padrão. Funciona com o mesmo princípio de antes, em que você envia dados para a pilha. Nesse caso, devemos colocar o nome da função primeiro, seguido por cada um dos argumentos em ordem. Uma vez que os tipos variam entre o C e Lua, você precisa utilizar a função correta para enviar o tipo apropriado para a pilha. Veja Listagem 6.

Isso teria o mesmo efeito que Lua chamando suas próprias funções; então,

```
result1, result2 =
swap_greeting("Steev", "Hello");
```

Observe que a ordem dos parâmetros é reversa, para combinar com sua remoção da pilha na ordem *First In, Last Out*.

Se quiséssemos os resultados dessa função no código em C, iríamos ler os dados da pilha e depois teríamos de removê-los explicitamente. Desta forma:

```
pResult2 = lua_tostring
(pLua, -1);
pResult1 = lua_tostring
(pLua, -2);
lua_pop(L, 2);
```

Repare na ordem reversa aqui também, e no uso de índices de pilha negativos.

Shepherd Moons

Em poucas páginas, conseguimos aprender os fundamentos de uma nova linguagem de programação, adicionar novos recursos a um navegador Web e encontrar uma maneira de incrementar nossos próprios projetos com scripts dinâmicos de configuração. Tudo isso graças ao poder e à flexibilidade de Lua. Estou certo de que você também já tem suas próprias idéias de projetos onde Lua pode ser útil. Vá em frente! ■

SOBRE O AUTOR

Quando operários vão a um bar, eles falam sobre futebol. Presume-se que, quando jogadores de futebol vão a um bar eles falam a respeito dos operários! Quando Steven Goodwin vai a um bar ele não fala sobre futebol ou operários. Ele fala sobre computadores. Invariavelmente...

Lua na terra dos mortos

Um dos maiores *cases* de sucesso no uso da linguagem Lua é o jogo *Grim Fandango*, um adventure da Lucas Arts. Segundo Bret Mogilefsky, programador chefe, originalmente o jogo deveria usar o *engine* SCUMM, o mesmo de todos os outros adventures da Lucas até a época. Contudo, SCUMM não era flexível o suficiente para as ambições do jogo e um novo engine teria de ser escrito. Procurando alternativas à tarefa de escrever um interpretador para uma nova linguagem de script, Bret encontrou Lua, na época na versão 2.50, que provou ser flexível e poderosa o suficiente para a tarefa, e acabou sendo usada em scripts para controlar diálogos e interações entre personagens, cenários e vários eventos do jogo. A linguagem também facilitou o processo de testes, já que no caso de um bug, os programadores podiam simplesmente chamar o interpretador Lua e fazer alterações no script em tempo real, sem ter de gerar um novo executável. Veja a entrevista [5] de Bret ao site grimfandango.net, onde ele conta mais detalhes sobre o uso de Lua.

INFORMAÇÕES

- [1] Usos de Lua:
<http://www.lua.org/uses.html>
- [2] Documentação de Lua:
<http://www.lua.org/manual/5.0/>
- [3] Fóruns sobre Lua:
<http://archive.neotonic.com/archive/lua-l>
- [4] Programa elinks:
<http://elinks.or.cz/download.html>
- [5] Entrevista com Bret Mogilefsky:
<http://www.grimfandango.net/?page=articles&pagenumber=2>

linuxUSER

Bem-vindo à LinuxUser!

Esta é uma seção especial dedicada a destacar programas úteis e interessantes para ajudá-lo no seu trabalho diário com o Linux no desktop. Aqui você encontrará informações sobre como utilizar programas comuns de forma mais eficiente, obterá um valioso embasamento técnico e conhecerá as últimas novidades em software para seu sistema operacional favorito.

Xandros Desktop 2.0187

Confira em nosso CD o sucessor do Corel Linux, considerado por muitos como uma das distribuições Linux mais fáceis de usar.

Gráficos em modo texto.....80

Acelere operações básicas de manipulação de imagens com o *ImageMagick*, um poderoso conjunto de utilitários baseado na linha de comando.

Seleção VI-sual.....82

Aurélio Marinho Jargas nos ensina a usar o modo de seleção visual do *vim*, facilitando a edição de textos.

É hora do show!.....84

Quem tem saudades do *Lotus ScreenCam* precisa conhecer o *XVidCap*, ferramenta essencial para quem quer capturar vídeos de ações em seu desktop.

Papo de Botequim87

Continuando nossa série, vamos aprender a lidar com o poderosíssimo *grep* e construir nosso primeiro programa, um catálogo para sua coleção de CDs.



Corel Linux retorna das cinzas

Xandros Desktop

Neste mês, trazemos em nosso CD a versão de livre circulação do Xandros

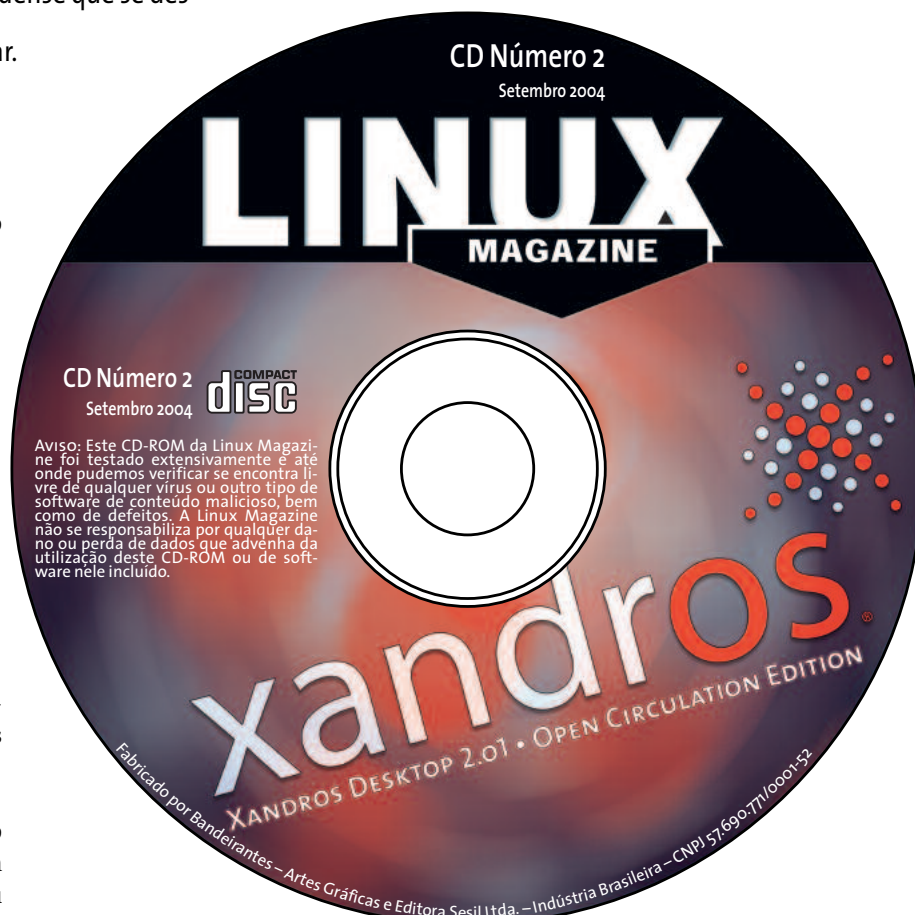
Desktop, uma distribuição Linux canadense que se destaca como uma das mais fáceis de usar.

POR RAFAEL RIGUES

Baseada na versão 3.0 do finado Corel Linux, esta distribuição tem como principais destaques a instalação e atualização simplificada de software, feita através da Internet pelo serviço Xandros Networks, e outras ferramentas como um poderoso gerenciador de arquivos capaz de gravar CDs, entre outros recursos, e ferramentas para facilitar a conexão à Internet.

A versão inclusa em nosso CD, a *Open Circulation Edition*, é uma espécie de “demonstração” das versões comerciais, *Xandros Desktop* e *Xandros Business Edition*. Embora completamente funcional, ela traz algumas poucas limitações:

- A velocidade de gravação de CDs no Xandros File Manager é limitada à velocidade mínima suportada pelo seu gravador. Programas externos, como o X-CD-Roast, não são afetados.
- Uma versão com banners de propaganda do navegador e cliente de e-mail Opera é instalada no sistema como navegador padrão. Você ainda tem a opção de instalar os aplicativos do projeto Mozilla, ou qualquer outro navegador e programa de e-mail que tenha um pacote para o Debian, gratuitamente, através do serviço/utilitário Xandros Networks.
- Não há guia do usuário, nem uma licença do CrossOver Office/Plugin, necessário para executar o MS Office e outros aplicativos Windows. Ambos podem ser adquiridos via Internet na Xandros Networks, ou obtidos ao atualizar sua distribuição para as versões Deluxe ou Business (ambas pagas).



- Esta versão não inclui suporte técnico, mas você pode tirar suas dúvidas nos fóruns no site da Xandros.

Instalação

Instalar o sistema é fácil. Na modalidade *Express Install* (Instalação Expressa) tudo o que você tem a fazer é iniciar o computador com o CD no drive e seguir as instruções na tela para criar um administrador do sistema, um usuário comum e aceitar o contrato de licença. Todo o resto é feito automaticamente para você. Usuários mais experientes podem escolher a modalidade *Custom Install* (Instalação Personalizada), que permite um ajuste fino do sistema, como a configuração da rede, seleção do software a ser instalado, particionamento manual do

disco, etc. Após o primeiro “boot”, um assistente guia o usuário pelos passos finais da configuração, como conexão à Internet, instalação da impressora, preferências regionais e aparência e comportamento do ambiente de trabalho.

INFORMAÇÕES

- [1] Xandros: <http://www.xandros.com>
- [2] Fóruns de Discussão: <http://www.xandros.com/support/forums.html>
- [3] Lista de Compatibilidade de Hardware: <http://www.xandros.com/support/forums.html>
- [4] ISO Language Codes: <http://www.w3.org/WAI/ER/IG/ert/iso639.htm>
- [5] ISO Country Codes: <http://www.iso.org/iso/en/prods-services/iso3166ma/o2iso-3166-code-lists/list-em.html>



Figura 1: Na opção Express Install basta criar alguns usuários...

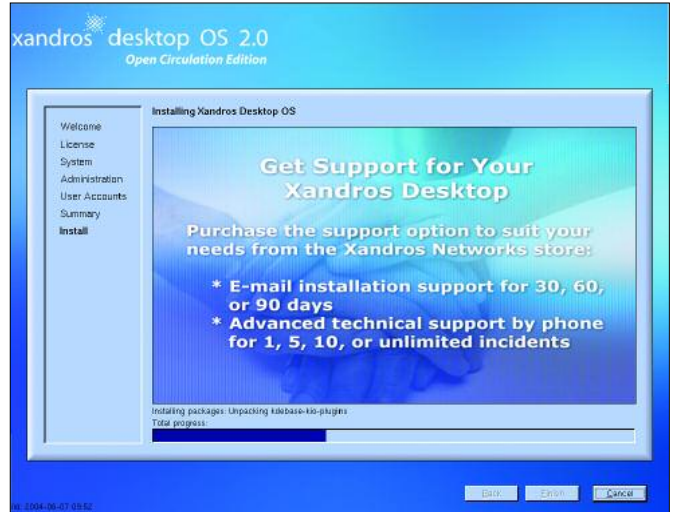


Figura 2: ...e aguardar a instalação do sistema.

O Peixe Babel

Oficialmente, o Xandros Desktop - Open Circulation Edition traz suporte a um único idioma, em duas versões: inglês norte-americano e britânico. Uma rápida busca na Xandros Networks revela a inexistência de pacotes oficiais com suporte a outros idiomas.

Mas existe um meio de adicionar “por debaixo dos panos” o suporte a quase qualquer um dos mais de 50 idiomas oficialmente suportado pelo KDE. Conecte-se à Internet e faça o seguinte:

- Abra a Xandros Networks (Launch ↵ Xandros Networks) e clique no menu Edit, item Set Applications Sources. Na janela que surge, marque a opção Debian unsupported site.
- Espere a atualização da lista de pacotes e, no campo de busca no topo da janela, digite kde-i18n-ptbr. Clique em Search.

- Em Search Results, na lateral esquerda da janela, clique no item Expert. Há um único resultado para nossa busca, justamente o pacote com a tradução do KDE e seus aplicativos para o Português do Brasil. Clique no link Install ao lado da descrição do pacote.
- Agora abra o editor de textos (Launch ↵ Applications ↵ Accessories ↵ Text Editor). Clique em File ↵ Open e em File Name digite .kde/share/config/kdeglobals e tecla Enter.
- Encontre a linha Language = en_US na seção Locale (geralmente é a linha 107 do arquivo) e substitua-a por: Language = pt_BR.
- Salve o arquivo (basta clicar no ícone do disquete) e reinicie o KDE.
- Pronto! Aplicativos da Xandros, como o Xandros Networks, continuarão em inglês, mas os aplicativos do KDE ficam em Português do Brasil.

Para instalar o suporte a outros idiomas, repita os passos acima, mas substitua pt_BR pelo código do idioma desejado, composto pelo ISO Language Code [4] e ISO Country Code [5]. O espanhol seria es_ES, japonês ja_JA, e assim por diante.

E com este truque você ainda ganha acesso a todos os pacotes no repositório apt do Debian. Experimente!

SOBRE O AUTOR

Rafael Rrigues, um maníaco por videogames e computadores antigos, já foi funcionário da Conectiva, onde participou do desenvolvimento do Conectiva Linux 5 e Conectiva Linux: e-Commerce, e membro da equipe da Revista do Linux, da qual foi editor durante um ano. Atualmente é um dos editores da Linux Magazine Brasil.


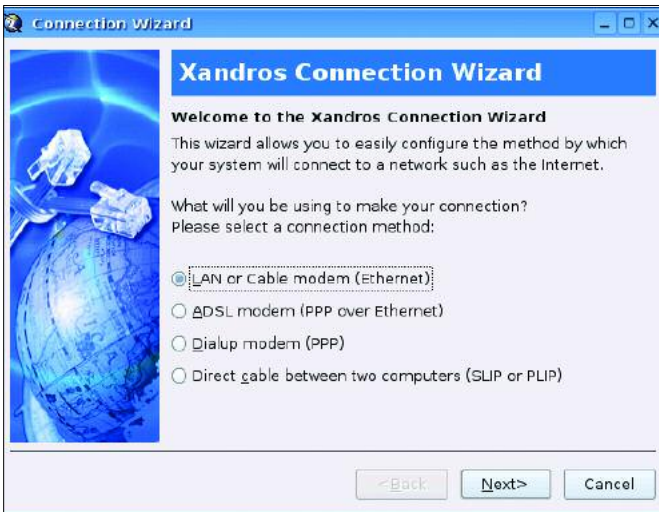



Figura 3: O assistente pós-instalação configura a conexão à internet...

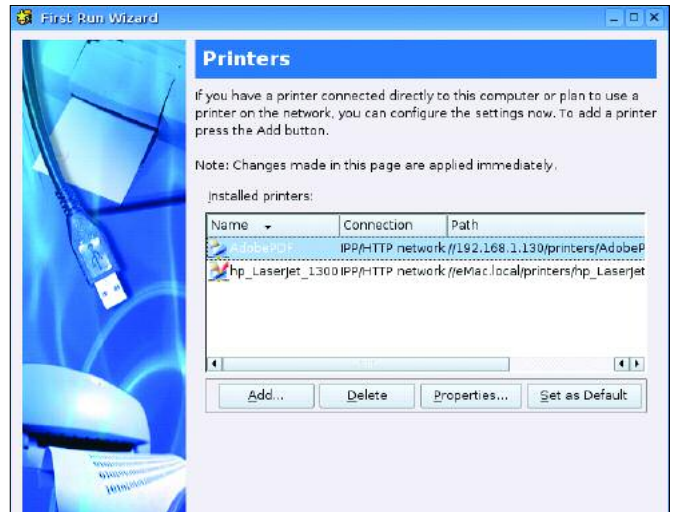
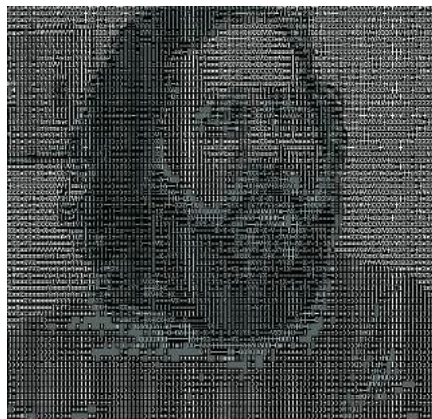


Figura 4: ...e suas impressoras, entre outros itens.

Ferramentas gráficas do ImageMagick

Gráficos em modo texto



Para a maioria das pessoas, manipulação de imagens é sinônimo de uma interface gráfica complexa, com inúmeras paletas cheias de ferramentas, seleções de cor e coisas do gênero. Infelizmente, existem muitas tarefas para as quais um programa como o Gimp é simplesmente inapropriado.

O ImageMagick [1], que muitas distribuições Linux instalam automaticamente, tem uma seleção de ferramentas em linha de comando para a manipulação de imagens. Trataremos de algumas delas nesse artigo.

Quadro 1: Script “color2gray”

```
01 #!/bin/bash
02 # Criando arquivos TIF e PDF
   monocromáticos
03 #
04 base=$(echo $1 | sed -e
   "s/\....$/") # Remove a
   extensão do arquivo
05 baseBW=${base}.tif
   # Nome do arquivo TIF
06 basePDF=${base}.pdf
   # Nome do arquivo PDF
07 convert -colors 256 -
   colorspace \
08 GRAY $1 $baseBW
   # Converte em TIF
09 convert $baseBW $basePDF
   # ... e em PDF
10 mkdir -p COLOR; mv $1 COLOR/
   # Altera original
```

O Gimp é o que a maioria das pessoas escolhe como aplicativo padrão para tarefas mais complexas de manipulação de imagem. Contudo, para muitas tarefas, como redimensionamento e conversão de formatos, a linha de comando oferece um caminho mais rápido. **POR HANS-GEORG ESSER**

Mudando Formatos

Converter imagens para outro formato, como TIF em PNG, é uma tarefa bastante comum. O comando do ImageMagick que faz isso é *convert*:

```
convert image.tif image.png
```

O *convert* avalia a extensão do arquivo em questão para definir o formato final. Não faz nenhuma diferença digitar a extensão em minúsculas ou maiúsculas, ou mesmo uma mistura das duas. A ferramenta lida até mesmo com variantes como *.tiff* ao invés vez de *.tif*, ou *.jpeg* ao invés de *.jpg*. Você pode digitar:

```
man ImageMagick
```

para abrir a página principal do manual (manpage) do ImageMagick e obter uma lista dos formatos de arquivo suportados.

Scripting

Uma das principais vantagens das ferramentas de linha de comando é o fato de poderem ser usadas em um script para automatizar as tarefas repetitivas.

Vamos imaginar que você precise converter regularmente screenshots a cores no formato PNG em arquivos TIF monocromáticos e em arquivos PDF em escala de cinza. Para fazê-lo, simplesmente use o script *color2gray* mostrado no Quadro 1. Ele não apenas converte o formato

PNG em TIF e PDF, como também reduz o universo de cores. As opções que manipulam isso são *-colors 256* e *-color space GRAY*.

O script também se encarrega de parte da organização do processo. Se você digitar o comando abaixo:

```
color2gray test.jpg
```

ele cria *test.tif* e *test.pdf* e move o arquivo original para um subdiretório chamado *COLOR* – se esse diretório não existir, o script simplesmente o cria.

Redimensionando

Na editora, precisamos ter gráficos de diferentes tamanhos da página de capa para cada edição (ver Figura 2).

O *convert* nos ajuda a fazer isso, usando a opção *-geometry* para mudar o tamanho da imagem. O tamanho final é especificado como *largura x altura*, como no comando:

```
convert big.png -geometry 150x200 small.png
```

para uma imagem pequena com uma largura até 150 pixels e uma altura até 200 pixels. Se você quer que o *convert* utilize exatamente esses valores (e distorça a imagem ao fazê-lo), tecle um sinal de exclamação imediatamente depois dos dados de tamanho:

Quadro 2: Script “conversão de documento”

```
01 #!/bin/sh
02 echo converting title: $1 to $2 \ (small) and $3 \ (very small)
03 convert $1 -blur 3 -geometry 220x312! $2
04 convert $1 -blur 3 -geometry 149x211! $3
```

```
convert big.png -geometry 2
150x200! small.png
```

Aqui na editora, usamos a listagem mostrada no Quadro 2 para preparar os gráficos da capa que serão publicados na Web. No nosso caso, faz sentido usar um tamanho fixo (ou seja, o parâmetro '!'); pois isso nos assegura que todas as miniaturas sejam do mesmo tamanho, além de nos permitir criar um layout com as miniaturas alinhadas uma ao lado da outra.

Além da conversão manual de arquivos, o *convert* também tem uma propriedade especial que cria um panorama com todas as imagens. Digite os seguintes comandos em um diretório contendo apenas imagens:

```
convert 'vid:*' directory.png
```

para criar um panorama com as miniaturas das imagens e os nomes dos arquivos, resolução e tamanho como legenda sobre fundo branco.

Você pode usar imagens desse tipo como “mapas de imagens clicáveis” em sites na Web. O utilitário *convert* pode ser usado para criar automaticamente destes mapas. O comando a seguir irá criar três arquivos:

```
convert 'vid:*' HTML:index.html
```

- *index.gif*: é a imagem em miniatura. O *convert* deduz o nome da imagem a partir do nome do arquivo HTML.

Quadro 3: Comando de Identificação

```
01 $ identify *.png *.jpg
02 john@linux:~/LMUK45/Cover> identify *.png *.jpg
03 ald15.png PNG 1704x1710+0+0 PseudoClass 256c 8-bit 877.4kb 0.150u 2
0:01
04 Cover_45.png[1] PNG 620x879+0+0 DirectClass 8-bit 345.0kb 0.040u 0:01
05 Cover_45s.png[2] PNG 220x312+0+0 DirectClass 8-bit 69.4kb 0.010u 0:01
06 Cover_45ss.png[3] PNG 149x211+0+0 DirectClass 8-bit 36.8kb 0.000u 2
0:01
07 directory.png[4] PNG 768x186+0+0 DirectClass 16-bit 268.6kb 0.020u 2
0:01
08 Manpage.png[5] PNG 636x966+0+0 DirectClass 8-bit 48.8kb 0.030u 0:01
09 little_duck.jpg[6] JPEG 800x600+0+0 DirectClass 8-bit 59.5kb 0.030u 2
0:01
```

Entrada e Saída

O *import* e o *display* são ferramentas bastante úteis. O primeiro cria screenshots, o outro mostra imagens. O *import* tem uma vasta gama de opções. Para capturar toda a área de trabalho, digite:

```
import -window root 2
desktop.png
```

E para capturar uma janela:

```
import -frame window.png
```

O parâmetro *-frame* é obrigatório, uma vez que, sem ele, o *import* remove a moldura

da janela e captura apenas o conteúdo.

Se você usa o ambiente de trabalho KDE, pode pressionar *[Alt + F2]* para abrir uma linha de comando, e experimentar usar o comando *import* em conjunto com o comando *sleep*:

```
sleep 3; import -window root 2
desktop.png
```

com isso você ganha algum tempo para organizar sua área de trabalho antes que o comando seja executado. Você pode, por exemplo, abrir alguns menus.

Finalmente, o *display* é um visualizador simples que mostra as imagens em seu tamanho real. Se não há espaço suficiente para isso, a ferramenta mostra uma janela de pré-visualização, permitindo que o usuário selecione as partes da imagem que lhe interessam.

Conclusão

convert, *identify*, *import* e *display* não são mais que uma pequena seleção do kit de ferramentas do ImageMagick e discutimos apenas superficialmente suas características. Para distribuições baseadas em RPM ou Debian, use os seguintes comandos para descobrir que programas estão incluídos no pacote:

```
rpm -qfl `which identify` | 2
grep bin
dpkg -L ImageMagick | grep bin
```

E lembre-se: quando tudo mais falhar, leia as páginas do manual! ■

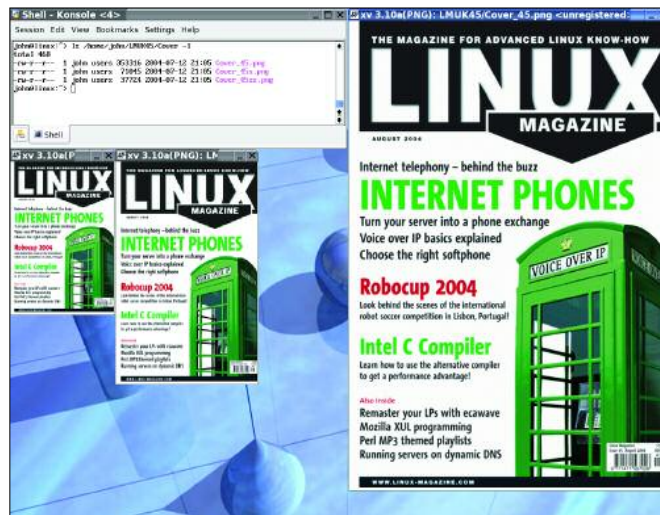


Figura 1: Usamos versões de diferentes tamanhos da imagem de capa na página da Linux Magazine.

- *index.html*: é um arquivo HTML com a imagem GIF embutida e devidamente mapeada; basta dar um clique na imagem em miniatura ser levado à imagem maior.
- *index_map.shtml*: contém a seção de mapeamento de *index.html* e é útil para integração (no servidor) com múltiplos sites na Web.

Detalhes

O comando *identify* também faz parte do pacote ImageMagick. Ele exibe informações sobre os arquivos que passam por ele, incluindo o tamanho em pixels (veja o Quadro 3) e a profundidade de cor (8 ou 16 bit no nosso exemplo).

O *identify* não utiliza a extensão do arquivo para identificar seu formato - mesmo se ela for removida, o utilitário continuará reconhecendo o arquivo. Ele indica o tamanho da imagem no formato *largura x altura*, assim como o *convert*.

Dominando o VI

Seleção VI-sual

O nome do VI vem da palavra “Visual”. Editor de textos revolucionário em sua época, sua primeira versão já trazia uma impressionante característica: podia mostrar na tela o texto que estava sendo editado!

AURÉLIO MARINHO JARGAS

Já pensou ter que editar um texto com o monitor desligado, só digitando os comandos e torcendo para que tudo fique certo? Pois assim eram os editores antes do VI, como o seu antecessor *ed*. Nada de cursor, setinhas, letras aparecendo e sumindo na tela. A edição era feita às escuras. E você achando que o VI é que é complicado, hein?

Pois bem, neste artigo vamos conhecer o “modo visual” do VI. O nome pode causar confusão, mas esse modo nada tem a ver com o fato do VI ser um editor visual. O modo visual é o nome que se dá quando fazemos uma seleção de texto, igual aos editores gráficos quando selecionamos um texto com o mouse.

Como o VI não depende de mouse para ser operado, há meios de selecionar um texto somente com o teclado. A tecla chave é o *V* (de Visual, claro!).

Diferente da seleção de texto em outros editores, que só serve para apagar o texto selecionado, copiá-lo ou trocá-lo

por outro texto, a seleção visual do VI é um aliado poderoso para fazer edições complicadas com pouco esforço.

Abra um texto qualquer no VI e brinque de selecionar trechos para se familiarizar com os três modos de seleção. Primeiro posicione o cursor no meio de uma linha, tecle *v* e mova o cursor com as setas do teclado para ver a seleção se formando: o texto selecionado ficará em destaque. Tecle *v* novamente (ou *Esc*) para cancelar a seleção.

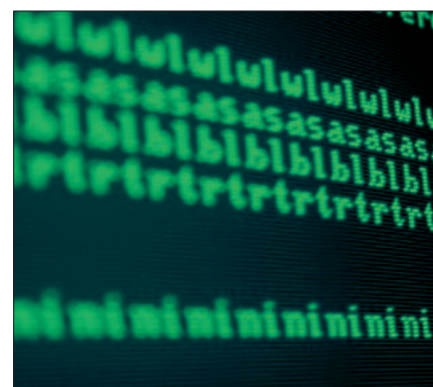
Agora volte ao mesmo meio da linha inicial e tecle *V* (maiúsculo). Surpresa! A linha inteira foi selecionada. Usando as setas para cima e para baixo podemos incluir mais linhas inteiras na seleção. Tecle *V* ou *Esc*, para cancelar a seleção.

Volte novamente ao meio da linha e tecle *Ctrl + V* para fazer a seleção de blocos. Mova o cursor para baixo e para a direita e veja um retângulo se formando. Antes de cancelar esta seleção, veja que é possível mudar o modo de seleção sem perder o posicionamento atual. Ainda com o bloco selecionado, tecle *v* para trocar para seleção de caracteres, depois tecle *V* para selecionar as linhas.

Tecele *Esc* para cancelar a seleção. Um comando bacana é o *gv* (*Go Visual*), que restaura a última seleção feita. Agora, com algumas linhas selecionadas, vamos fazer algo com elas. Veja na tabela 2 alguns dos comandos disponíveis.

Variantes do modo visual

Comando	Nome	Ação
<i>v</i>	--VISUAL--	Seleciona caracteres
<i>V</i>	--VISUAL LINE--	Seleciona linhas
<i>Ctrl+V</i>	--VISUAL BLOCK--	Seleciona blocos (seleção vertical)



Charles Wilson - www.sxc.hu

Brinque bastante com cada um desses comandos para se acostumar e veja que seu uso do VI ficará muito mais ágil. Por exemplo, quer apagar todo um conjunto de linhas? Vá até a primeira, aperte *V* e desça com o cursor até a última linha, então aperte *d* e pronto: linhas apagadas.

O comando *r* é interessante no modo de seleção: ele troca todos os caracteres do trecho selecionado por um caractere à sua escolha, bom para “esconder” uma parte do texto.

Muito úteis também são o *u* e o *U*, para deixar o texto todo em minúsculas ou maiúsculas, não importando como era o trecho original. Experimente também o *gq*, que reformata as linhas, juntando-as e dividindo novamente para que a largura máxima (*:set textwidth*) seja respeitada.

E por último, o poder dos comandos “!” e “:” aplicados à seleção. Para ordenar alfabeticamente apenas algumas linhas do texto, e não ele todo, selecione as linhas desejadas com o *V* e digite *!sort* seguido de *Enter*. Para substituir palavras apenas na seleção, novamente selecione o texto desejado e tecele *:s/isso/aquilo/* e tecele *Enter*.

Ao usar esses comandos, surge na última linha da tela o texto *:’<,’>*. Não se preocupe, ele está ali para indicar que o comando digitado em seguida será aplicado apenas ao trecho selecionado.

Note que após aplicado o comando, o VI volta para o modo anterior, abandonando o modo visual. Resumindo, esse modo é composto por três operações:

1. Iniciar o modo visual com *v*, *V* ou *Ctrl + V*
2. Mover o cursor para selecionar o texto
3. Aplicar um comando à seleção

Alguns comandos do VI

Comando	Ação no texto selecionado
<i>d</i>	Apaga
<i>y</i>	Copia
<i>p</i>	Cola
<i>c</i>	Troca por outro texto
<i>r</i>	Troca cada letra por um caractere
<i>J</i>	Junta todas as linhas em uma só
<i>U</i>	Converte para maiúsculas
<i>u</i>	Converte para minúsculas
<i>~</i>	Inverte maiúsculas e minúsculas
<i>gq</i>	Reformata as linhas para a largura desejada.
<i>!</i>	Manda as linhas para um comando externo (<i>!sort</i>)
<i>:</i>	Aplica um comando ‘ex’ nas linhas (<i>:s/isso/aquilo/</i>)

Durante o passo 2 de demarcar a seleção, se você apertar a tecla *o* (letra *o* de ontem, minúscula), o cursor alterna entre o início e o fim do trecho marcado. Dessa maneira é possível aumentar o tamanho de uma seleção já iniciada.

Isso é tudo o que qualquer usuário normal precisa saber para usar e abusar da seleção visual e tornar a edição de textos no vi mais confortável. A partir daqui entraremos num mundo bizarro, onde coisas estranhas e surpreendentes acontecem. Continue lendo por sua conta e risco.

A seleção de blocos (*Ctrl + V*) reserva algumas surpresas. Para entender o que vai acontecer com os comandos a seguir, é imprescindível que você tenha o VI aberto com um texto qualquer e repita exatamente em sua máquina todos os procedimentos descritos.

Já precisou incluir no início de algumas linhas um texto padrão? É possível fazer isso com um `:s/^/TEXTO/`, mas com a seleção visual isso é muito mais divertido. Vá com o cursor até o início de uma linha qualquer, aperte *Ctrl + V* e desça sete linhas com a seta. Tecla *I*

(maiúsculo) e logo em seguida digite *MÁGICA*. Agora pare, respire fundo, olhe atentamente para a tela e tecla *Esc*. Meio segundo se passa e *PLIM!*, letras pipocam na tela. Isso é mágica!

Este foi o comando *I*, que Insere um texto antes da seleção. Para inserir depois, o comando é *A*, de Anexar. Este fica de exercício para você brincar sozinho. O comando que fecha a trinca é o *c* (minúsculo), de *Change*, que troca o texto selecionado por outro.

O comando *c* é muito útil para editar tabelas. Selecione uma coluna vertical no meio da linha com *Ctrl + V*, digamos cinco linhas e cinco colunas. Tecla *c*, depois digite *@@@@@*. Novamente respire fundo, tecla *Esc* e *PLIM!*, a mágica acontece!

A última dica da seleção em bloco é útil para quando o bloco precisa ir até o final de todas as linhas, mas estas têm tamanhos diversos. Nesse caso o bloco sairá incompleto se a última linha não for a mais longa. Enquanto estiver movendo o cursor para marcar o bloco, aperte *\$* para expandir o bloco até o final de todas as linhas selecionadas.

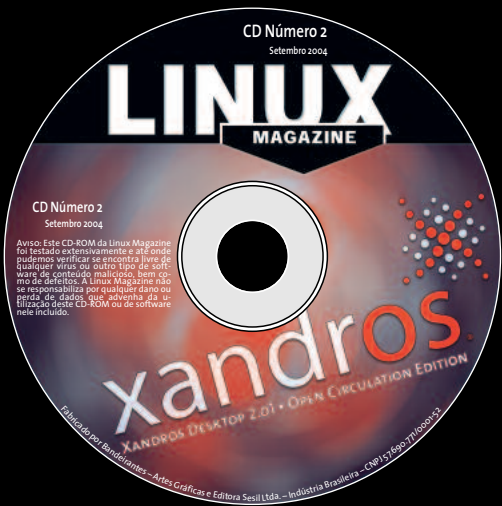
Ah! E antes tarde do que nunca, um aviso: o modo visual não faz parte do VI original, é uma característica presente apenas no Vim, o *Vi Improved*. Como ele é o “Vi” padrão da maioria das distros recentes, a seleção visual está disponível a praticamente todos os usuários. ■

INFORMAÇÕES

- [1] <http://www.vim.org>
- [2] <http://www.aurelio.net/>
- [3] <http://www.cs.pdx.edu/~kirkenda/joy&4.html>

SOBRE O AUTOR

Aurélio Marinho Jargas é pesquisador e programador, especialista em Linux e com experiência em Bash, Sed, PHP e Python. Sua especialidade é automação, manipulação de texto, formatação e conversões entre tipos diferentes de dados. É o autor das funções ZZ e do primeiro depurador (“debugger”) feito para a linguagem Sed (sedsed), além de um livro sobre expressões regulares publicado pela Editora Novatec.



CD-ROM da Linux Magazine Não saia de casa sem ele!

E no fim do ano, todos os artigos da revista em um único CD-ROM, com máquina de busca para você achar rapidinho tudo o que precisa!

Central de Assinaturas:

Linux New Media do Brasil
 Av. Luís Carlos Berrini, 1500
 Cj. 103 – Brooklin Novo
 São Paulo – SP – Brasil
 Tel.: 0xx11 3345 1002
 Fax: 0xx11 3345 1081
assinaturas@linuxnewmedia.com.br

- Todo mês um CD-ROM diferente
- Repleto de programas interessantes
- Coletâneas especiais para facilitar a sua vida
- Distribuições Linux prontas para instalar

Capture vídeos em seu desktop

É hora do show!

Às vezes, pode ser um tanto complicado descrever as funções de um aplicativo apenas com palavras. Um screenshot pode ajudar, mas se você precisa de mais do que uma figura estática, o *xvidcap* pode ser a resposta.

POR ANDREA MÜLLER

Gravar os passos de programas da área de trabalho para fins de demonstração ou treinamento nunca foi uma tarefa fácil. Mesmo que dúzias de programas estejam competindo pela atenção de usuários que produzem documentação, nenhum deles é de fato muito bom com imagens. Descubra o *xvidcap* [1] escrito por Rasca Gmelch. O programa, mantido e desenvolvido por Karl H. Beckers, foi feito na medida para criar demonstrações de ações na tela. Pode criar imagens ou arquivos MPEG, definíveis pelo usuário, da área de trabalho, sendo assim perfeitamente adequado para mostrar o caminho através de estruturas de menu e caixas de diálogo.

As Novas Ferramentas do Diretor

Antes de dar seus primeiros passos como um diretor de filmes, você obviamente precisa instalar o programa. Isso é bastante fácil para usuários cuja distribuição é compatível com os pacotes RPM ou Debian da homepage do *xvidcap*. O RPM fornecido pelo desenvolvedor do programa roda no Suse Linux 9.0 e Red Hat 9.0, entre outros. Há dois pacotes para o Debian: um para a versão atual (*Woody*) e outro para a versão de teste, mais conhecida como *Sarge*.

Usuários de outras distribuições precisarão compilar eles mesmos o *xvidcap*. Isso quer dizer que você deve ter em seu sistema os pacotes de desenvolvimento

da *libpng*, *libjpeg*, e *XFree86* instalados. Apenas siga o seguinte procedimento padrão: `./configure ; make ; su -c make install` para compilar o *xvidcap* e instalar o programa no diretório `/usr/local`. Como o *root* é o único usuário com permissão para escrever nesse nível, você deve entrar com `su -c` para temporariamente assumir os privilégios de *root* no passo `make install`.

O *xvidcap* depende do programa *ffmpeg* [2] para criar vídeos. Se ele não está instalado no seu sistema, a sua versão do *xvidcap* ainda poderá criar screenshots, mas não vídeos. Se a sua distribuição não inclui o pacote *ffmpeg*, será necessário compilar a versão *ffmpeg* incluída na distribuição do *xvidcap*. Para isso, adicione o parâmetro `--with-forced-embedded-ffmpeg` ao comando `./confi-`

`gure`, para criar um *xvidcap* com *ffmpeg* embutido (*embedded*). Uma segunda opção é mudar para o subdiretório *ffmpeg* antes de compilar o *xvidcap*. Entre com `./configure; make; su -c make install` nesse diretório para criar o codificador *ffmpeg* e o player *ffplay* e depois digite `make install` para instalar essas ferramentas em `/usr/local/bin`. A vantagem desse segundo procedimento é que ele permite utilizar o *ffmpeg* e o *ffplay* independentemente do *xvidcap*, o que a versão “embutida” não possibilita.

A opção `--with-gtk2 ./configure` é também interessante se você mesmo estiver compilando o programa, já que ele cria um programa chamado *gvidcap*. Diferente do original, o *gvidcap* tem uma moderna interface baseada na biblioteca *Gtk 2* que combina perfeitamente com a atual área de trabalho do *Gnome*.

O Script

Para iniciar a nova ferramenta instalada no passo anterior, digite um dos comandos abaixo:

```
xvidcap &
gvidcap &
```



Figura 1: A interface tradicional do *xvidcap*.



Figura 2: A versão *Gtk 2* tem uma aparência mais moderna que o original.

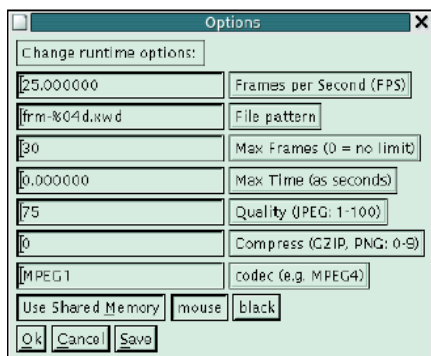


Figura 3: Use esse diálogo para especificar o formato de saída e o nome do arquivo de saída.

O primeiro comando inicia a interface baseada em Motif, e o segundo a baseada em Gtk 2, mais “moderna”. Os elementos de controle para o *xvidcap* (ver Figura 1) e o *gvidcap* (ver Figura 2) são organizados exatamente da mesma forma. Isso traz uma vantagem e uma desvantagem. A versão original parece um pouco tediosa, mas certamente é mais estável e muito mais configurável do que a versão mais atual. Estabilidade, configurabilidade e o fato de que ambas as interfaces são similares, são o que nos fizeram decidir concentrar-nos no *xvidcap* neste artigo.

Quando você inicia o programa, aparece na tela uma moldura vermelha do tamanho de um cartão de crédito para indicar a área de gravação. A moldura move-se com a janela do programa. Para evitar isso, você pode clicar no botão x para separar a moldura do resto do aplicativo. Como o tamanho padrão é pequeno demais para qualquer coisa, mesmo para a menor das ferramentas, é melhor definir a área de trabalho que você quer gravar antes de fazer qualquer outra coisa. Clique na mira da barra de ferramentas para transformar o cursor do mouse em uma mira. Você pode também segurar o botão esquerdo do mouse e arrastá-lo para definir uma área, ou cli-

car numa janela, se você estiver interessado apenas no conteúdo. Ao selecionar uma janela, o videoclipe não irá incluir sua barra de título.

Depois de selecionar o local, o próximo passo é definir alguns parâmetros, como o formato da saída. [Alt + O] (ou [Ctrl + P] para o *gvidcap*) irá abrir a caixa de configuração mostrada na Figura 3.

Antes de tudo, defina os valor de *Max Frames* em 0, já que o *xvidcap* tem uma restrição padrão de 30 imagens. O valor 0 remove esse limite, permitindo que você crie vídeos de qualquer duração. Use a caixa *Frames per Second (FPS)* para especificar o número de imagens por segundo que o *xvidcap* irá capturar. O número padrão é 25 - não fique confuso pela maneira como ele é mostrado (25.000000). Se você precisa criar vídeos em MPEG1 ou MPEG4, veja a página de manual do *xvidcap* (option `—fps`) para os valores necessários de FPS. Por exemplo, 25 e 50 são velocidades válidas para MPEG1, enquanto o MPEG4 pode usar valores entre 7.5 e 30. Se você entrar com um valor inválido, seus vídeos não vão rodar ou muito rápido ou muito devagar.

Coloque a codificação do vídeo no campo *codec* (por exemplo, *MPEG4*). O padrão é *MPEG1*, mas você pode mudá-lo para *MPEG4* e ter uma qualidade de gravação superior. A resolução dos vídeos MPEG1 que nós criamos era tão pequena que, às vezes, não conseguíamos decifrar alguns menus.

Vídeo ou Imagem?

Você precisa ainda digitar o nome do arquivo de saída na caixa *File pattern*. Se você deixar o nome padrão, *frm-%04d.xwd*, o *xvidcap* não irá criar um vídeo, mas, ao invés disso, irá gerar arquivos no formato *xwd* (*x window dump*) no diretório onde você iniciou o

programa. A cadeia de caracteres *%04d* adiciona um número serial ao nome do arquivo da imagem, criando os arquivos *frm-0000.xwd* até *frm-9999.xwd*. O sinal de porcentagem mostra informações da formatação: *04d* (*d* para decimal) especifica o número de dígitos. Se sua demonstração tem mais de 10.000 imagens, simplesmente mude o valor de *File pattern* para *frm-%05d.xwd*, ou mesmo *frm-%06d.xwd*. Você pode modificar tanto o nome do arquivo de base (*frm*) como o formato de saída. O *xvidcap* também pode criar arquivos JPEG ou PNG. Imagens isoladas são muito úteis se você precisa editar o resultado final ou se está tentando obter um screenshot complexo, como uma mensagem de erro que aparece em sua tela e desaparece, imediatamente, que programas como o *Ksnapshot* não conseguem capturar.

Para criar um vídeo diretamente, mude a entrada em *File pattern* para qualquer nome de arquivo usando *mpeg* como extensão, por exemplo *program-demo.mpeg*. Para evitar ter de modificar as opções toda vez que você usa o programa, clique em *Save*. Isso diz ao *xvidcap* para guardar suas modificações no arquivo *.xvidcap.scf* dentro do seu diretório pessoal. O companheiro em *Gtk 2* do *xvidcap*, o *gvidcap*, lê o mesmo arquivo. Em outras palavras, qualquer mudança que você fizer em um programa irá imediatamente afetar o outro. Apague o *~/.xvidcap.scf* para recuperar as configurações padrão do programa.

Tomada Um, Ação!

Clique no botão com o ponto vermelho para começar a gravar a área de trabalho dentro da moldura vermelha. Depois de mostrar os aspectos do software que você quer, clique no botão com o quadrado preto para parar de gravar. Você pode usar um player como o *MPlayer*

GLOSSÁRIO

MPEG: Abreviação de *Motion Picture Expert Group*. Grupo que engloba empresas e universidades e padronizou técnicas de compressão para formatos de vídeo. A codificação MPEG tenta mostrar o máximo possível de segmentos de imagem ressaltando como elas se diferem da imagem anterior, e permite criar arquivos de menor tamanho. Há várias versões do formato MPEG, como MPEG-1 (VCD), MPEG-2 (DVD), MPEG-4 (DVD) e MPEG-4 Parte 10 (ou H.264/AVC, para dispositivos móveis).

GTK 2: “The Gimp Toolkit” (Conjunto de Ferramentas do Gimp), versão 2. Esta biblioteca gráfica fornece aos desenvolvedores ferramentas prontas para a criação de interfaces gráficas, como botões, barras de rolagem, caixas de diálogo e campos de entrada. Além do Gnome, XFCE e muitos outros softwares, o Gimp 2 também baseado na GTK 2. Sua principal concorrente é a Qt (pronuncia-se “qjutie”, da norueguesa TrollTech, usada no ambiente de trabalho KDE).

ImageMagick: Coleção de dez ferramentas de manipulação de imagens. A maior parte dessas ferramentas é baseada em linha de comando, tornando o ImageMagick perfeito para integração em Shell Scripts.

~/.Xdefaults: Arquivo onde os usuários definem as preferências para programas com interface gráfica, como a cor de fundo ou a fonte. Os programas mais recentes do KDE e do Gnome ignoram as mudanças feitas em *~/.Xdefaults*.

(veja Figura 4) para exibir sua demonstração para o público. Para simplificar as coisas, você pode optar por usar a linha de comando de execução de vídeo *ffplay* que acompanha o *ffmpeg*. O seguinte comando inicia a ferramenta

```
ffplay videodemo.mpeg
```

Pressionar *[q]* encerra a apresentação.

Depois de criar uma série de imagens, você pode clicar nos botões com as setas para percorrer as imagens individualmente. Clique no botão com a imagem de um lápis, à direita para abrir a imagem atual no *Display*, o visualizador de imagens do pacote **ImageMagick**. Você pode então clicar na janela da imagem para abrir um menu com uma série de funções de manipulação de imagem (veja Figura 5) como mudanças de cor e tamanho (*Enhance*), rotação (*Transform*) e uma série de efeitos (*F/X*).

O botão com o desenho da tira de filme chama-se *animate*, outra ferramenta da coleção do **ImageMagick**. Como sugere o nome, o *animate* cria e executa uma animação feita com os arquivos das imagens individuais. Evite usar essa opção com um grupo de mais de 400 imagens. Fornecer muito material ao *animate* irá derrubar até mesmo os sistemas mais poderosos.

De Imagens a Videoclipes

O *File* \rightarrow *Make Video* converte suas imagens em um videoclipe. O *xvidcap* examina o arquivo *~/Xdefaults* para

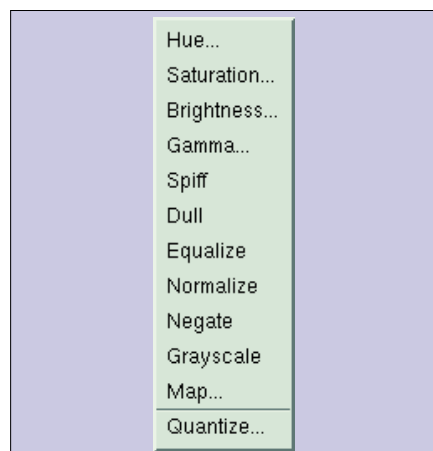


Figura 5: o *xvidcap* usa o utilitário *Display* para abrir imagens individuais. Ele tem uma série de funções para manipulação de imagens, como o menu *Enhance*, com opções para modificar o brilho, valores de gama e saturação.

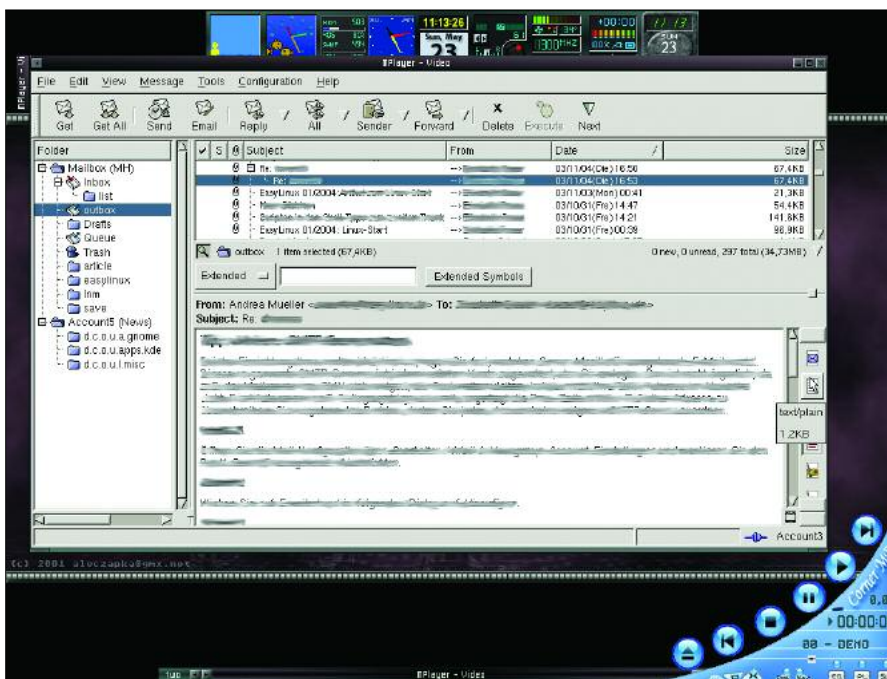


Figura 4: O *MPlayer* executando um vídeo criado com o *xvidcap* sobre leitores de e-mail.

descobrir que programa deve ser usado para essa tarefa. É aqui que os padrões dos programas gráficos ficam armazenados. O *gvidcap* não honra o *~/Xdefaults*, então o comando de conversão só funciona para o *xvidcap*. Vamos fazer de conta que você tem uma coleção de imagens no formato PNG, com os nomes baseados em uma cadeia de caracteres como *pic%04d.png* e quer usar o *ffmpeg* para converter essas imagens em um videoclipe MPEG1. Adicione a seguinte linha ao arquivo *~/Xdefaults*:

```
xvidcap*mkVideoCommand:ffmpeg
-i %s video.mpeg
```

Ela diz ao *xvidcap* para chamar o *ffmpeg* quando você seleciona o comando *Make Video*. Para que isso funcione, você deve usar o *ffmpeg* propriamente dito, não é possível usar a versão embutida no *xvidcap*. O *xvidcap* substitui a cadeia de caracteres *%s* pelo padrão de nome usados nas imagens individuais, em nosso exemplo *pic%04d.png*. Como o *ffmpeg* reconhece essa nomenclatura, cria arquivos sequenciais chamados *pic0000.png*, *pic0001.png* e salva o arquivo de saída, *video.mpeg*, sob o diretório atual.

Há uma armadilha ao utilizar o *ffmpeg* para gerar um vídeo a partir de arquivos de imagens individuais. O codificador não sabe lidar com os arquivos *xwd* que o *xvidcap* cria como padrão. O *ffmpeg*

utiliza os seguintes formatos de imagem: *pnm*, *pam*, *pgmyuv*, *yuv*, *png*, *jpeg*, e *gif*. O formato PNG é uma boa escolha ao capturar screenshots com o *xvidcap*. Para selecioná-lo, informe a extensão do arquivo como *.PNG* no item *File pattern* no menu de opções do *xvidcap*.

O caminho mais rápido para o sucesso para diretores amadores é dizer ao *xvidcap* para criar uma saída de vídeo e deixar o processamento de imagens individuais para usuários que estão insatisfeitos com videoclipes medianos e querem modificar o material bruto antes de gerar o videoclipe. ■

INFORMAÇÕES

- [1] Página oficial do *xvidcap*:
<http://sourceforge.net/projects/xvidcap>
- [2] Projeto *ffmpeg*:
<http://ffmpeg.sourceforge.net/>

SOBRE O AUTOR

Após quase dois anos como jornalista independente, Andrea Mueller agora trabalha como editora para a *Linux New Media AG*.



Quando não está lidando com artigos ou empacotando software para a área de downloads da revista *EasyLinux*, ela gosta de ir além do *Linux*, e se envolve com outros sistemas operacionais, como o *QNX*, *BeOS* and *NetBSD*.

Curso de Shell Script

Papo de Botequim - Parte II

Nossos personagens voltam à mesa do bar para discutir expressões regulares e colocar a “mão na massa” pela primeira vez, construindo um aplicativo simples para catalogar uma coleção de CDs. **POR JÚLIO CÉSAR NEVES**

Garçom! Traz um “chops” e dois “pastel”. O meu amigo hoje não vai beber porque está finalmente sendo apresentado a um verdadeiro sistema operacional, e ainda tem muita coisa a aprender!

- E então, amigo, tá entendendo tudo que te expliquei até agora?
- Entendendo eu tô, mas não vi nada prático nisso...
- Calma rapaz, o que te falei até agora serve como base ao que há de vir daqui pra frente. Vamos usar essas ferramentas que vimos para montar programas estruturados. Você verá porque até na TV já teve programa chamado “O Shell é o Limite”. Para começar vamos falar dos comandos da família *grep*
- *Grep*? Não conheço nenhum termo em inglês com este nome...
- É claro, *grep* é um acrônimo (sigla) para *Global Regular Expression Print*, que usa expressões regulares para pesquisar a ocorrência de cadeias de caracteres na entrada definida.

Por falar em expressões regulares (ou *regex*), o Aurélio Marinho Jargas escreveu dois artigos [1 e 2] imperdíveis para a Revista do Linux sobre esse assunto e também publicou um livro [3] pela Editora Novatec. Acho bom você ler esses artigos, eles vão te ajudar no que está para vir.

Eu fico com *grep*, você com gripe

Esse negócio de gripe é brincadeira, só um pretexto para pedir umas caipirinhas. Eu te falei que o *grep* procura cadeias de caracteres dentro de uma entrada definida, mas o que vem a ser uma “entrada definida”? Bem, existem várias formas de definir a entrada do comando *grep*. Veja só. Para pesquisar em um arquivo:



```
$ grep franklin /etc/passwd
```

Pesquisando em vários arquivos:

```
$ grep grep *.sh
```

Pesquisando na saída de um comando:

```
$ who | grep carvalho
```

No 1º exemplo, procurei a palavra *franklin* em qualquer lugar do arquivo */etc/passwd*. Se quisesse procurar um nome de usuário, isto é, somente no início dos registros desse arquivo, poderia digitar `$ grep '^franklin' /etc/passwd`.

“E para que servem o circunflexo e os apóstrofos?”, você vai me perguntar. Se tivesse lido os artigos que mencionei, saberia que o circunflexo serve para limitar a pesquisa ao início de cada linha e os apóstrofos servem para o Shell não interpretar esse circunflexo, deixando-o passar incólume para o comando *grep*.

No 2º exemplo mandei listar todas as

linhas que usavam a palavra *grep*, em todos os arquivos terminados em *.sh*. Como uso essa extensão para definir meus arquivos com programas em Shell, malandramente, o que fiz foi listar as linhas dos programas que poderia usar como exemplo do comando *grep*.

Olha que legal! O *grep* aceita como entrada a saída de outro comando, redirecionado por um *pipe* (isso é muito comum em Shell e é um tremendo acelerador da execução de comandos). Dessa forma, no 3º exemplo, o comando *who* listou as pessoas “logadas” na mesma máquina que você (não se esqueça jamais: o Linux é multiusuário) e o *grep* foi usado para verificar se o Carvalho estava trabalhando ou “çoçando”.

O *grep* é um comando muito conhecido, pois é usado com muita frequência. O que muitas pessoas não sabem é que existem três comandos na família *grep*: *grep*, *egrep* e *fgrep*. A principais diferenças entre os 3 são:

- *grep* - Pode ou não usar expressões regulares simples, porém no caso de não usá-las, o *fgrep* é melhor, por ser mais rápido.
- *egrep* (“e” de *extended*, estendido) - É muito poderoso no uso de expressões regulares. Por ser o mais poderoso dos três, só deve ser usado quando for necessária a elaboração de uma expressão regular não aceita pelo *grep*.
- *fgrep* (“f” de *fast*, rápido) - Como o nome diz, é o ligeirinho da família, executando o serviço de forma muito veloz (por vezes é cerca de 30% mais rápido que o *grep* e 50% mais que o *egrep*), porém não permite o uso de expressões regulares na pesquisa.

-Agora que você já conhece as diferenças entre os membros da família, me diga: o que você acha dos três exemplos que eu dei antes das explicações?

- Achei que o *fgrep* resolveria o teu problema mais rapidamente que o *grep*.
- Perfeito! Tô vendo que você está atento, entendendo tudo que estou te explicando! Vamos ver mais exemplos

Quadro 1 - Listando subdiretórios

```
$ ls -l | grep '^d'
```

drwxr-xr-x	3	root	root	4096	Dec 18	2000	doc
drwxr-xr-x	11	root	root	4096	Jul 13	18:58	freeciv
drwxr-xr-x	3	root	root	4096	Oct 17	2000	gimp
drwxr-xr-x	3	root	root	4096	Aug 8	2000	gnome
drwxr-xr-x	2	root	root	4096	Aug 8	2000	idl
drwxrwxr-x	14	root	root	4096	Jul 13	18:58	locale
drwxrwxr-x	12	root	root	4096	Jan 14	2000	lyx
drwxrwxr-x	3	root	root	4096	Jan 17	2000	pixmap
drwxr-xr-x	3	root	root	4096	Jul 2	20:30	scribus
drwxrwxr-x	3	root	root	4096	Jan 17	2000	sounds
drwxr-xr-x	3	root	root	4096	Dec 18	2000	xine
drwxr-xr-x	3	root	root	4096	Jun 19	2000	xpls

para clarear de vez as diferenças de uso entre os membros da família.

Eu sei que em um arquivo qualquer existe um texto falando sobre Linux, só não tenho certeza se está escrito com *L* maiúsculo ou minúsculo. Posso fazer uma busca de duas formas:

```
egrep (Linux | linux) arquivo.txt
```

ou então:

```
grep [Ll]linux arquivo.txt
```

No primeiro caso, a expressão regular complexa (*Linux* | *linux*) usa os parênteses para agrupar as opções e a barra vertical (|) é usada como um “ou” (*or*, em inglês) lógico, isto é, estou procurando *Linux* ou *linux*.

No segundo, a expressão regular *[Ll]linux* significa: começado por *L* ou *l* seguido de *linux*. Como esta é uma expressão simples, o *grep* consegue resolvê-la, por isso é melhor usar a segunda forma, já que o *egrep* tornaria a pesquisa mais lenta.

Outro exemplo. Para listar todos os subdiretórios do diretório corrente, basta usar o comando `$ ls -l | grep '^d'`. Veja o resultado no Quadro 1.

No exemplo, o circunflexo (^) serviu para limitar a pesquisa à primeira posição da saída do *ls* longo (parâmetro

-l). Os apóstrofes foram usados para o Shell não “ver” o circunflexo. Vamos ver mais um. Veja na Tabela 1 as quatro primeiras posições possíveis da saída de um *ls -l* em um arquivo comum (não é diretório, nem link, nem ...).

Para descobrir todos os arquivos executáveis em um determinado diretório eu poderia fazer:

```
$ ls -la | egrep '^-..(x|s)'
```

novamente usamos o circunflexo para limitar a pesquisa ao início de cada linha, ou seja, listamos as linhas que começam por um traço (-), seguido de qualquer coisa (o ponto), novamente seguido de qualquer coisa, e por fim um *x* ou um *s*. Obteríamos o mesmo resultado se usássemos o comando:

```
$ ls -la | grep '^-..[xs]'
```

e além disso, agilizaríamos a pesquisa.

A “CDteca”

Vamos começar a desenvolver programas! Creio que a montagem de um banco de dados de músicas é bacana para efeito didático (e útil nestes tempos de downloads de arquivos MP3 e queimadores de CDs). Não se esqueça que, da mesma forma que vamos desenvolver um monte de programas para organizar os seus CDs de música, com pequenas adaptações você pode fazer o mesmo para organizar os CDs de software que vêm com a Linux Magazine e outros que você compra ou queima, e disponibilizar esse banco de software para todos os que trabalham com você (o Linux é multiusuário, e como tal deve

ser explorado).

– Péra aí! De onde eu vou receber os dados dos CDs?

– Vou mostrar como o programa pode receber parâmetros de quem o estiver executando e, em breve, ensinarei a ler os dados da tela ou de um arquivo.

Passando parâmetros

Veja abaixo a estrutura do arquivo contendo a lista das músicas:

```
nomedoálbum^intérprete1~nome?
damúsical:...:intérpreten~nome?
damúsican
```

Isto é, o nome do álbum será separado por um circunflexo do resto do registro, formado por diversos grupos compostos pelo intérprete de cada música do CD e a música interpretada. Estes grupos são separados entre si por dois pontos (:) e, internamente, o intérprete será separado por um til (~) do nome da música.

Quero escrever um programa chamado *musinc*, que incluirá registros no meu arquivo músicas. Passarei cada álbum como parâmetro para o programa:

```
$ musinc “álbum^interprete~?
musica:interprete~musica:...”
```

Desta forma, *musinc* estará recebendo os dados de cada álbum como se fosse uma variável. A única diferença entre um parâmetro recebido e uma variável é que os primeiros recebem nomes numéricos (o que quis dizer é que seus nomes são formados somente por um algarismo, isto é, \$1, \$2, \$3, ..., \$9). Vamos, fazer mais alguns testes:

```
$ cat teste
#!/bin/bash
#Teste de passagem de parametros
echo “1o. parm -> $1”
echo “2o. parm -> $2”
echo “3o. parm -> $3”
```

Agora vamos rodar esse programinha:

```
$ teste passando parametros para ?
testar
bash: teste: cannot execute
```

Ops! Esqueci-me de tornar o script executável. Vou fazer isso e testar novamente o programa:

Tabela 1

Posição	Valores possíveis
1ª	-
2ª	r ou -
3ª	w ou -
4ª	x,s(suid) ou -

```
$ chmod 755 teste
$ teste passando parametros para 3
testar
1o. parm -> passando
2o. parm -> parametros
3o. parm -> para
```

Repare que a palavra *testar*, que seria o quarto parâmetro, não foi listada. Isso ocorreu porque o programa *teste* só lista os três primeiros parâmetros recebidos. Vamos executá-lo de outra forma:

```
$ teste "passando parametros" 3
para testar
1o. parm -> passando parametros
2o. parm -> para
3o. parm -> testar
```

As aspas não deixaram o Shell ver o espaço em branco entre as duas primeiras palavras, e elas foram consideradas como um único parâmetro. E falando em passagem de parâmetros, uma dica: veja na Tabela 2 algumas variáveis especiais. Vamos alterar o programa *teste* para usar as novas variáveis:

```
$ cat teste
#!/bin/bash
# Programa para testar passagem
de parametros (2a. Versao)
echo 0 programa $0 recebeu $# 3
parametros
echo "1o. parm -> $1"
echo "2o. parm -> $2"
echo "3o. parm -> $3"
echo Para listar todos de uma 3
\"tacada\" eu faco $*
```

Listagem 1: Incluindo CDs na "CDteca"

```
$ cat musinc
#!/bin/bash
# Cadastra CDs (versao 1)
#
echo $1 >> musicas
```

Listagem 2

```
$ cat musinc
#!/bin/bash
# Cadastra CDs (versao 2)
#
echo $1 >> musicas
sort -o musicas musicas
```

Execute o programa:

```
$ teste passando parametros para testar
0 programa teste recebeu 4 3
parametros
1o. parm -> passando
2o. parm -> parametros
3o. parm -> para
Para listar todos de uma 3
"tacada" eu faco passando 3
parametros para testar
```

Repare que antes das aspas usei uma barra invertida, para escondê-las da interpretação do Shell (se não usasse as contrabarras as aspas não apareceriam).

Como disse, os parâmetros recebem números de 1 a 9, mas isso não significa que não posso usar mais de nove parâmetros. Significa que só posso endereçar nove. Vamos testar isso:

```
$ cat teste
#!/bin/bash
# Programa para testar passagem
de parametros (3a. Versao)
echo 0 programa $0 recebeu $# 3
parametros
echo "1lo. parm -> $11"
shift
echo "2o. parm -> $1"
shift 2
echo "4o. parm -> $1"
```

Execute o programa:

```
$ teste passando parametros para 3
testar
0 programa teste recebeu 4 3
parametros que são:
1lo. parm -> passando1
2o. parm -> parametros
4o. parm -> testar
```

Duas coisas muito interessantes aconteceram neste script. Para mostrar que os nomes dos parâmetros variam de \$1 a \$9 digitei *echo \$11* e o que aconteceu? O Shell interpretou como sendo \$1 seguido do algarismo 1 e listou *passando1*;

O comando *shift*, cuja sintaxe é *shift n*, podendo o *n* assumir qualquer valor numérico, despreza os *n* primeiros parâmetros, tornando o parâmetro de ordem *n + 1*.

Bem, agora que você já sabe sobre passagem de parâmetros, vamos voltar à nossa "cdteca" para fazer o script de

inclusão de CDs no meu banco chamado *musicas*. O programa é muito simples (como tudo em Shell). Veja a Listagem 1.

O script é simples e funcional; limito-me a anexar ao fim do arquivo *musicas* o parâmetro recebido. Vamos cadastrar 3 álbuns para ver se funciona (para não ficar "enchendo lingüiça," suponho que em cada CD só existem duas músicas):

```
$ musinc "album3^Artista5^
~Musica5:Artista6~Musica5"
$ musinc "album1^Artista1^
~Musical:Artista2~Musica2"
$ musinc "album 2^Artista3^
~Musica3:Artista4~Musica4"
```

Listando o conteúdo do arquivo *musicas*:

```
$ cat musicas
album3^Artista5~Musica5:Artista6^
~Musica6
album1^Artista1~Musical:Artista2^
~Musica2
album2^Artista3~Musica3:Artista4^
~Musica4
```

Podia ter ficado melhor. Os álbuns estão fora de ordem, dificultando a pesquisa. Vamos alterar nosso script e depois testá-lo novamente. Veja a listagem 2. Simplesmente inseri uma linha que classifica o arquivo *musicas*, redirecionando a saída para ele mesmo (para isso serve a opção *-o*), após cada álbum ser anexado.

```
$ cat musicas
album1^Artista1~Musical:Artista2^
~Musica2
albu2^Artista3~Musica3:Artista4^
~Musica4
album3^Artista5~Musica5:Artista6^
~Musica6
```

Oba! Agora o programa está legal e quase funcional. Ficará muito melhor em uma nova versão, que desenvolveremos após aprender a adquirir os dados da tela e formatar a entrada.

Tabela 2: Variáveis especiais

Variável	Significado
\$0	Contém o nome do programa
\$#	Contém a quantidade de parâmetros passados
\$*	Contém o conjunto de todos os parâmetros (muito parecido com @\$)

Ficar listando arquivos com o comando *cat* não está com nada, vamos fazer um programa chamado *muslist* para listar um álbum, cujo nome será passado como parâmetro. Veja o código na Listagem 3:

Vamos executá-lo, procurando pelo álbum 2. Como já vimos antes, para passar a string *album 2* é necessário protegê-la da interpretação do Shell, para que ele não a interprete como dois parâmetros. Exemplo:

```
$ muslist "album 2"
grep: can't open 2
musicas: album1^Artista1~Musical2
:Artista2~Musica2
musicas: album2^Artista3~Musica3
:Artista4~Musica4
musicas:album3^Artista5~Musica5
:Artista6~Musica6
```

Que lambança! Onde está o erro? Eu tive o cuidado de colocar o parâmetro passado entre aspas para o Shell não o dividir em dois! É, mas repare como o *grep* está sendo executado:

```
grep $1 musicas
```

Mesmo colocando *álbum 2* entre aspas, para que fosse encarado como um único parâmetro, quando o *\$1* foi passado pelo Shell para o comando *grep*, transformou-se em dois argumentos. Dessa forma, o conteúdo da linha que o *grep* executou foi o seguinte:

```
grep album 2 musicas
```

Como a sintaxe do *grep* é:

Listagem 3 - muslist

```
$ cat muslist
#!/bin/bash
# Consulta CDs (versao 1)
#
grep $1 musicas
```

Listagem 4 muslist melhorado

```
$ cat muslist
#!/bin/bash
# Consulta CDs (versao 2)
#
grep -i "$1" musicas
```

```
grep <cadeia de caracteres> [arq1, arq2, ..., arqn]
```

O *grep* entendeu que deveria procurar a cadeia de caracteres *album* nos arquivos 2 e *musicas*. Como o arquivo 2 não existe, *grep* gerou o erro e, por encontrar a palavra *album* em todos os registros de *musicas*, listou a todos.

É melhor ignorarmos maiúsculas e minúsculas na pesquisa. Resolveremos os dois problemas com a Listagem 4.

Nesse caso, usamos a opção *-i* do *grep* que, como já vimos, serve para ignorar maiúsculas e minúsculas, e colocamos o *\$1* entre aspas, para que o *grep* continuasse a ver a cadeia de caracteres resultante da expansão da linha pelo Shell como um único argumento de pesquisa.

```
$ muslist "album 2"
album2^Artista3~Musica3:Artista4
~Musica4
```

Agora repare que o *grep* localiza a cadeia pesquisada em qualquer lugar do registro; então, da forma que estamos fazendo, podemos pesquisar por álbum, por música, por intérprete e mais. Quando conhecermos os comandos condicionais, montaremos uma nova versão de *muslist* que permitirá especificar por qual campo pesquisar.

Ah! Em um dia de verão você foi à praia, esqueceu os CDs no carro, aquele "solzinho" de 40 graus empenou seu disco favorito e agora você precisa de uma ferramenta para removê-lo do banco de dados? Não tem problema, vamos desenvolver um script chamado *musexc*, para excluir estes CDs.

Antes de desenvolver o "bacalho", quero te apresentar a uma opção bastante útil da família de comandos *grep*. É a opção *-v*, que quando usada lista todos os registros da entrada, exceto o(s) localizado(s) pelo comando. Exemplos:

```
$ grep -v "album 2" musicas
album1^Artista1~Musical:Artista2
~Musica2
album3^Artista5~Musica5:Artista6
~Musica6
```

Conforme expliquei antes, o *grep* do exemplo listou todos os registros de *musicas* exceto o referente a *album 2*, porque atendia ao argumento do co-

Listagem 5 - musexc

```
$ cat musexc
#!/bin/bash
# Exclui CDs (versao 1)
#
grep -v "$1" musicas > /tmp/mus$$
mv -f /tmp/mus$$ musicas
```

mando. Estamos então prontos para desenvolver o script para remover CDs empenados da sua "CDteca". Veja o código da Listagem 5.

Na primeira linha mandei para */tmp/mus\$\$* o arquivo *musicas*, sem os registros que atendessem a consulta feita pelo comando *grep*. Em seguida, movi */tmp/mus\$\$* por cima do antigo *musicas*. Usei o arquivo */tmp/mus\$\$* como arquivo de trabalho porque, como já havia citado no artigo anterior, o *\$\$* contém o PID (identificação do processo) e, dessa forma, cada um que editar o arquivo *musicas* o fará em um arquivo de trabalho diferente, evitando colisões.

Os programas que fizemos até aqui ainda são muito simples, devido à falta de ferramentas que ainda temos. Mas é bom praticar os exemplos dados porque, eu prometo, chegaremos a desenvolver um sistema bacana para controle dos seus CDs. Na próxima vez que nos encontrarmos, vou te ensinar como funcionam os comandos condicionais e aprimoraremos mais um pouco esses scripts. Por hoje chega! Já falei demais e estou de goela seca! Garçom! Mais um sem colarinho! ■

INFORMAÇÕES

- [1] <http://www.revistadoinux.com.br/ed/003/ferramentas.php3>
- [2] <http://www.revistadoinux.com.br/ed/007/ereg.php3>
- [3] <http://www.aurelio.net/er/livro/>

SOBRE O AUTOR

Julio Cezar Neves é Analista de Suporte de Sistemas desde 1969 e trabalha com Unix desde 1980, quando fez parte da equipe que desenvolveu o SOX, um sistema operacional similar ao Unix, produzido pela Cobra Computadores.



Trabalho em conjunto para o bem da comunidade

Unindo esforços

Este mês vamos falar sobre os grupos de usuários e sua importância para a comunidade. Discutiremos também o movimento Open Source e suas principais diferenças em relação ao movimento do Software Livre.



O movimento do Software Livre só ganhou amplitude graças aos grupos de usuários, entidades compostas por indivíduos que se unem em torno de uma causa em comum, trocam informações, definem projetos e marcam encontros para atrair outros interessados na causa.

A Comunidade Sol [1], uma ONG de Manaus/AM que tem como objetivo fomentar projetos de Software Livre, uniu-se ao Debian-AM [2], grupo de usuários especializado na distribuição Debian, para organizar o Dia D, encontro que reuniu cerca de 500 participantes, mais do que o esperado pela organização. O encontro foi composto por palestras, um laboratório de instalação e outras atividades. Entre os participantes, era possível encontrar desde pessoas que nunca utilizaram Software Livre até desenvolvedores experientes.

A Comunidade Sol e o Debian-AM são exemplos de grupos regionais que ganharam o respeito do público de todas as idades e todos os níveis de envolvimento com Software Livre, graças à competência de seus organizadores. O mesmo ocorre em todos os cantos do planeta. No site do Projeto GNU [3] você encontrará links para os principais grupos seguem a filosofia do Software Livre.

Os grupos que ainda não estão listados no site podem se cadastrar seguindo as instruções na própria página. Vale lembrar que o cadastro de grupos denominados "Linux User Group" (Grupo de Usuários Linux), "LUG" (ou a variante em português, GUL) ou similares não é aceito, pois os mantenedores do site acreditam que a utilização dessa nomenclatura causa confusão e não repassa os devidos créditos ao Projeto GNU. Em caso de dúvidas, basta entrar em contato

via e-mail. O endereço está no site.

É comum ver pessoas que ingressam em grupos de usuários para aprender mais e se tornam membros da comunidade. Se observarmos quem desempenha algum papel em projetos de Software Livre, vamos notar que a grande maioria iniciou seus trabalhos em grupos de usuários. Enfim, Software Livre é colaboração, é liberdade de conhecimento e, quanto mais praticarmos isto, mais temos a ganhar.

Existem grupos de usuários sobre os mais variados temas, alguns bem específicos, como o projeto PythonBrasil [4], para os amantes da linguagem de programação Python. As mulheres também se organizaram para fundar o Projeto Software Livre Mulheres [5], onde as integrantes se reúnem para discutir os mais variados assuntos relacionados ao Software Livre junto com todo o charme feminino. É uma forma encontrada pelas fundadoras do grupo para incentivar as mulheres a participar mais ativamente de projetos de Software Livre. Um movimento ainda mais antigo é o Linux Chix Brasil. Mulheres, seja bem-vindas!

Open Source e Software Livre

É muito comum as pessoas associarem os termos Open Source e Software Livre como sendo a mesma coisa. A utilização de um nome ou termo diferente está diretamente relacionada a uma idéia diferente, exatamente o que acontece com os dois movimentos.

Em 1998 alguns participantes da comunidade do Software Livre começaram a utilizar o termo Open Source ao invés de Software Livre para descrever o que faziam. Naturalmente o termo Open Source passou a ser visto como uma idéia diferente, uma filosofia

diferente. Apesar das diferenças ideológicas, os dois grupos trabalham em conjunto em muitos projetos essenciais.

A principal diferença entre os dois movimentos está ligada a valores, na maneira como enxergam o mundo. Para o movimento Open Source, o fato de um software ser aberto é uma questão de praticidade, não uma questão ética. Para o Software Livre, essa é uma questão fundamentalmente social. Apesar de serem movimentos diferentes, os grupos não são inimigos; apenas pensam de forma diferente. O inimigo nesse caso é o software proprietário.

Outro fator que provavelmente contribuiu para a criação do movimento Open Source foi a ambigüidade existente no idioma inglês em relação ao termo "Free Software". O termo "free" pode ser interpretado tanto como "grátis" quanto como "livre", como na frase "liberdade de conhecimento". ■

INFORMAÇÕES

[1] <http://www.comunidadesol.org>

[2] <http://www.debian-am.org>

[3] <http://www.gnu.org/gnu/gnu-user-groups.html>

[4] <http://www.pythonbrasil.com.br>

[5] <http://mulheres.softwarelivre.org>

[6] <http://www.linuxchix.org.br>

SOBRE O AUTOR

Christiano Anderson (anderson@gnu.org) é desenvolvedor autônomo de Software Livre, participa do Projeto Software Livre Brasil (PSL-BR) e contribui com o Projeto GNU. Sua principal missão é difundir o software livre e sua filosofia, principalmente na educação.



Eventos em Destaque

Fórum Regional de Software Livre do ABCD**Data:** 21 a 25 de Setembro**Local:** Santo André, São Paulo**Website:**

forum.publicolivre.org.br

II CONISLI**Data:** 05 a 06 de Novembro**Local:** São Paulo, São Paulo**Website:**

www.conisli.org.br

Futurecom 2004**Data:** 25 a 28 de Outubro**Local:** Florianópolis, Santa Catarina**Website:**

www.futurecom.com.br

Calendário de Eventos

Evento	Data	Local	Website
2ª Semana de Software Livre da UERJ	10 de Outubro	Rio de Janeiro, RJ	www.2semana.linuerj.uerj.br
2º Congresso Catarinense de Software Livre	08 de outubro	Joinville, SC	www.softwarelivresc.org.br
2º CONISLI	05 e 06 de novembro	São Paulo, SP	www.conisli.org.br
Latinoware 2004	08 a 12 de novembro	Foz do Iguaçu, PR	www.latinoware.pr.gov.br
LinuxWorld Conference & Expo 2004	26 a 28 de novembro	Frankfurt, Alemanha	www.linuxworldexpo.de
I Fórum Empresarial de TI	18 a 22 de Outubro	Fernandópolis, SP	www.fef.br
Futurecom 2004	25 a 28 de Outubro	Florianópolis, SC	www.futurecom.com.br

Índice de Anunciantes

Anunciante	Site	Página
Alternativa Linux	www.alternativainux.com.br	75
Consist	www.consist.com.br	99 (terceira capa)
CD-ROM da Linux Magazine	www.linuxmagazine.com.br/Subs	83
FASP	www.fasp.br	65
Green Treinamento e Consultoria	www.green.com.br	11
Infomedia	www.infomediatv.com.br	91
iSolve	www.isolve.com.br	53
LinuxPress	www.linuxpress.com.br	09
Latinoware 2004	www.latinoware.pr.gov.br	55
Mais Informática	www.mav.com.br	69
MySQL Brasil	www.mysqlbrasil.com.br	45
Oracle	www.oracle.com.br	02 (contra capa)
Semana do CPPF de Tecnologia da Informação	forum.publicolivre.org.br	33
Sun Microsystems	www.sun.com.br	100 (quarta capa)

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 TIF e EPS.

Uma revista passa por muitos estágios durante sua produção, portanto alguns meses podem se passar desde que seu artigo seja recebido até que a revista chegue às bancas. Portanto, nunca envie na última hora material ou notícias sobre encontros e eventos.

Envie suas colaborações para o endereço material@linuxmagazine.com.br. Evite enviar mais de 4 MB em arquivos anexos. Caso o material para seu artigo ultrapasse este limite, coloque-o em algum site na internet e nos informe a URL.

Expediente Editorial

Editores

Rafael Antonio Guido Peregrino da Silva:
rperegrino@linuxmagazine.com.br
Rafael Pereira Rigues:
rrigues@linuxmagazine.com.br

Centros de Competência

Centro de Competência em Software:
Patricia Jung, pjung@linux-magazine.com
Centro de Competência em Hardware:
Mirko Dölle, mdoelle@linux-magazine.com
Centro de Competência em Redes e Segurança:
Achim Leitner, aleitner@linux-magazine.com

Correspondentes & Colaboradores

Alexandre Barbosa, Andrea Müller, Aurélio Marinho Jargas, Bruno Bellantuono, Carlos Morimoto, Carsten Wiese, Carsten Schnober, Charly Kühnast, Christiano Anderson, Daniel Trezub, Hans-Georg Esser, Jan Kleinert, Jörg Reitter, Júlio César Neves, Marc André Selig, Marcos Vinícius Pecy Marini, Maurício Soares dos Anjos, Mirko Dölle, Piter Punk, Rafael Peregrino da Silva, Rafael Pereira Rigues, Steven Goodwin, Sulamita Garcia, Telsa Gwynne, Tim Schürmann, Thomas Wölfer, Zack Brown.

Revisão

Henrique César Ulbrich:
henrique.cesar@ermida.com
Julia Vidili:
julia.vidili@ermida.com

Direção de Arte

Luciano Hagge Dias:
lhagge@linuxmagazine.com.br

Design da Capa

Gisele Sarian:
gisele-c@ig.com.br
David Ballot:
davidballot@ig.com.br
Gaveteiro: Montarq Services
www.montarq.com.br

Anúncios: www.linuxmagazine.com.br/Advertise
Brasil

Unical - União Comercial e Editora Ltda.
A/c. Sérgio Alday
anuncios@linuxmagazine.com.br
Tel./Fax: +55 (0)11 3051 3153
Cel.: +55 (0)11 9935 1368

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 Antonio Guido Peregrino da Silva,
rperegrino@linuxmagazine.com.br

Linux Magazine

Av. Eng.º Luiz Carlos Berrini, 1500
Conj. 103 - Brooklin Novo
04571-000 - São Paulo - SP - Brasil
Tel.: +55 (0)11 3345 1002
Fax: +55 (0)11 3345 1081

Assinaturas: www.linuxmagazine.com.br/Subs

Preço: (12 edições incluindo CD mensal): R\$125,60
Email: assinaturas@linuxmagazine.com.br

Preço Unitário: R\$12,90

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.co.uk - Reino Unido
www.linux-magazine.pl - Polônia
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 advinham 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. Assuma-se que qualquer correspondência recebida, tal como cartas, emails, fax, 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: Gráfica Bandeirantes

ISSN 1806-9428

Impresso no Brasil



em processo de filiação

Os Monges DANIEL TREZUB



Terceira Edição

No próximo mês...

Voice Over IP – A revolução

Usar seu computador como um telefone já foi um sonho distante. Mas com o rápido aumento no poder de processamento das modernas CPUs e a queda nos custos de acesso à Internet, o sonho está, aos poucos, se tornando realidade para muitas pessoas e empresas.

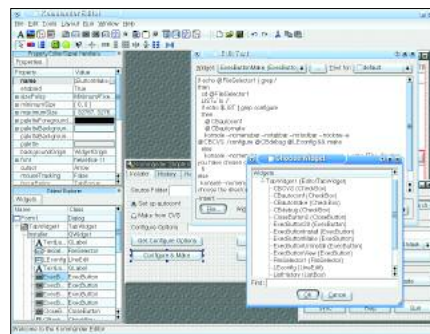
Nossa matéria de capa começa com uma visão geral sobre os conceitos básicos de Voice Over IP (VoIP). Começamos mostrando como você pode integrar VoIP a seus sistemas já existentes, e do que você precisa para ser capaz de fazer e receber chamadas para qualquer um que use um telefone normal e o sistema de telefonia atual. Forneceremos uma visão geral dos codecs mais usados pelas operadoras de gateways VoIP, testaremos os sete *softphones* mais populares no mundo Linux, comparando seus recursos e funções, e mostraremos como controlar sua nova central telefônica com o *Asterisk*, desde a instalação até todos detalhes da configuração.



Os poderes do KDE

Esqueça os ícones Crystal por um instante, existem mais coisas interessantes por detrás das cortinas do KDE. Este excelente ambiente de trabalho possui alguns recursos chave que em conjunto permitem que você faça praticamente qualquer coisa dentro dele de forma automatizada, e nesse texto daremos uma olhada em três deles. Eis a trupe: DCOP, KIO e o caçula Kommander, uma espécie de versão anabolizada do Dialog que pode ser usado para adicionar janelas e caixas de diálogo no padrão do KDE a seus scripts.

Você aprenderá a usar o DCOP (*Desktop Communication Protocol*) para interagir com aplicativos do KDE já abertos, o KIO para acessar servidores de e-mail, CDs de áudio, listas de fontes e preferências do sistema e verá como criar scripts simples com uma bela interface com o Kommander.



As novidades no Gnome 2.8

Sayamindu Dasgupta nos leva a uma jornada através de novos recursos do Gnome 2.8, lançado em 15 de Setembro de 2004.

Este popular ambiente de trabalho para o Linux, cada vez mais refinado, agora conta com um cliente e-mail com filtro integrado (*Novell Evolution*) e um novo sistema de arquivos a partir de um novo gerenciador de *applets* para o painel, suporte à tecnologia *DNS-*



G · N · O · M · E

SD, popularizada pela Apple com *iDev*, um conjunto de ferramentas independentes, independentes, para configuração de aspectos de sua Linux chamado *tem Tools*, componente de desktop através do cliente VNC (o melhor integrado) e detecção de hardware, principalmente câmeras digitais e dispositivos USB com o HAL (*Hardware Abstraction Layer*) e muitos outros recursos.

Linux Magazine Preview

A Linux Magazine Preview é uma newsletter via email que lhe oferece uma espiadinha no conteúdo da próxima edição, além de novidades sobre material on-line disponibilizado em nosso website, antes que a revista chegue às bancas. Para receber, cadastre-se aqui: www.linuxmagazine.com.br/Newsletter

Em breve nas bancas!

CD Número 2

Setembro 2004

LINUX

MAGAZINE

CD Número 2 COMPACT
Setembro 2004 **disc**

Aviso: Este CD-ROM da Linux Magazine foi testado extensivamente e até onde pudemos verificar se encontra livre de qualquer vírus ou outro tipo de software de conteúdo malicioso, bem como de defeitos. A Linux Magazine não se responsabiliza por qualquer dano ou perda de dados que advinha da utilização deste CD-ROM ou de software nele incluído.

xandros[®]
XANDROS DESKTOP 2.01 • OPEN CIRCULATION EDITION

Fabricado por Bandeirantes - Artes Gráficas e Editora Sesil Ltda. - Indústria Brasileira - CNPJ 57.699.771/0001-52