



Emily Stone: National Science Foundation - EUA

Linux e seus sabores

Por que existem tantas distribuições? Parte II

Na edição passada, acompanhamos a história e principais características do Debian e do

Fedora. Nesta edição, é a vez do Gentoo e do Slackware. **POR SULAMITA GARCIA**

O Gentoo [1] foi criado por Daniel Robbins, que começou usando Debian e criou uma versão desta distribuição chamada Enoch. O objetivo de Robbins era tornar o Enoch uma distro muito rápida, que tivesse criação e atualização de pacotes completamente automatizadas. Com o tempo, o Enoch se distanciou tanto do Debian que se tornou o Gentoo. Na época em que o Gentoo chegaria à versão 1.0, Robbins adquiriu uma máquina que tinha uma incompatibilidade de hardware com o Linux. Por causa disso, o projeto ficou parado por um tempo. Nesse período, Robbins começou a usar o FreeBSD e se tornou fã do ports, um repositório de pacotes extremamente eficiente, que compila os pacotes na máquina por padrão antes de instalá-los. Essa experiência serviu de inspiração para o Portage, repositório de pacotes do Gentoo. Logo depois o problema com o hardware foi consertado e Daniel voltou ao Linux. Com a ajuda de outros desenvolvedores, como Achim Gottinger, o Gentoo voltou à ativa.

O Gentoo possui um *Contrato Social* baseado no do Debian, porém simplificado. Esse contrato traz basicamente as mesmas premissas do documento que o inspirou: “O Gentoo é e permanecerá Software Livre”, “o código será devolvido à comunidade” e “não esconderemos os problemas”. Atualmente a distribuição tem suporte às arquiteturas x86, AMD64, PowerPC, UltraSparc, Alpha and MIPS. Possui mais de 8000 pacotes no seu repositório, o Portage, e está na versão 2004.3.

Uma definição aproximada do Gentoo seria dizer que ele é um *Linux From Scratch* automatizado. A documentação disponível no site é a melhor que já vi, extremamente didática e detalhada. Usuários que queiram começar a usar o Linux, mesmo que com outra distribuição, irão se beneficiar com os tópicos sobre particionamento, sistemas de arquivos e formatação de discos, configuração e conceitos básicos de rede, além de compreender as variáveis envolvidas na otimização de um sistema, como por exemplo a definição

de otimizações adequadas ao processador da máquina. Essa documentação foi recentemente atualizada, em 17 de janeiro, com uma versão voltada para a instalação a partir do terceiro estágio (*stage 3*) de um Live CD. O grupo GentooBR [2] lançou há pouco tempo uma versão em português dessa documentação. Vale a pena conferir e também colaborar com outros projetos do grupo.

A distribuição concentra-se na otimização do desempenho do sistema,



Figura 1: O Gentoo é ideal para quem quer o máximo de desempenho do sistema.

usando para isso muita automatização. O processo gira em torno da configuração de parâmetros no arquivo `/etc/make.conf` através das variáveis `CXXFLAGS` e `USE`. Os `CXXFLAGS` são parâmetros passados ao `gcc`. Com eles é possível selecionar, entre outras coisas, seu processador. Como este artigo trata de características gerais das distribuições, não iremos nos aprofundar nisso. A variável `USE`, por outro lado, irá definir como serão compilados os pacotes a instalar. Há uma longa lista de recursos para ativar ou desativar ao gosto do freguês, entre eles: Java, ALSA, criptografia, X, QT e GTK. Assim, esses recursos serão automaticamente ativados ou desativados, de acordo com suas escolhas, durante a compilação. Obviamente, se um pacote não tiver, por exemplo, suporte à criptografia, ativá-lo ou desativá-lo não terá efeito algum.

A instalação do Gentoo se divide em estágios – os *stages*. Eles definem de onde você vai começar a instalação. É possível começar diretamente do *stage 3* se você quiser, porém a diversão está em começar pelo menos do *stage 2*.

No *stage 1*, você irá executar o script `bootstrap.sh`, que irá compilar a `libc`, o `gcc` e vários outros programas essenciais do sistema, todos otimizados para seu processador. Isso demora bastante.

O *stage 2* é a compilação e instalação do sistema básico, de acordo com as configurações escolhidas ao se configurar a variável `USE`. Você pode baixar um arquivo de *stage 2* já preparado para seu processador nos mirrors do Gentoo e assim pular a parte do bootstrap. No *stage 3*, você pode escolher entre o kernel da série 2.4 ou 2.6 e se quer um kernel puro ou se gostaria de usar um conjunto de patches de otimização. E todo o resto, interface gráfica, clientes de irc, icq, serviços de rede...

Tudo isso é feito através do *emerge*, a ferramenta que resolve as dependências, verifica os recursos ativados na variável `USE` e instala os pacotes de acordo com as opções ali listadas. Através dele é possível atualizar toda a listagem do sistema e os pacotes, com os comandos `emerge sync` e `emerge world`. Com o *emerge*, também se podem baixar pacotes binários, o que às vezes é aconselhável, como no caso do OpenOffice. Os pacotes são constantemente atualizados,

coisa que pode ser problemática: por ser uma distro preocupada com otimização e constante atualização, o Gentoo não é lá extremamente estável. Li num site que isso se deve a um pobre sistema de QA – *Quality Assurance* (Controle de Qualidade) – que idealmente deveria pegar os bugs antes do sistema ser liberado. Passei por problemas com a versão 2004.2 em um Athlon XP instalado a partir de um *stage 1* e só consegui resolvê-los usando o *stage 2*.

A instalação é dolorida – nunca espere gastar menos de alguns dias para ter um sistema funcional. Você pode gerar pacotes com o software que está instalando para reaproveitar mais tarde, sem ter de compilar tudo novamente. Porém, dois sistemas com algumas poucas configurações diferentes não irão aproveitar os pacotes uma da outra. Por exemplo, um usuário que goste de GTK e Python não vai aproveitar os pacotes compilados com suporte a QT e Java.

Todavia, após alguns dias de instalação, o resultado é impressionante. Foi emocionante ver o Mozilla e o OpenOffice.org nem sequer pensarem para abrir, numa máquina em que a única coisa não compilada era o próprio OpenOffice. Se você gosta de emoções fortes, de controlar à unha seu sistema e fazê-lo render ao máximo, porém com ferramentas automatizadas e scripts de inicialização prontos, o Gentoo é uma ótima pedida. Pode parecer incrível, mas existem pessoas, como uma colega minha, que quando não têm o que compilar buscam os últimos pacotes adicionados ao Portage só para ver seu sistema em ação. Há loucos pra tudo. Se você é um desses, leia o Gentoo Handbook [9] e divirta-se.

Slackware

O nascimento do Slackware [3] já foi contado na primeira edição desta revista. Resumidamente, ele é a mais antiga distribuição Linux em atividade, criada por Patrick Volkerding. Seu criador atualmente enfrenta sérios problemas de saúde, porém conta com o apoio e trabalho de vários grupos ao redor do mundo, liderados pelo GUS-BR [4], na manutenção do sistema e na torcida pelo seu pronto restabelecimento.

Na primeira vez que ouvi falar do Slackware, soube que era uma distribuição para hackers. Um pouco mais

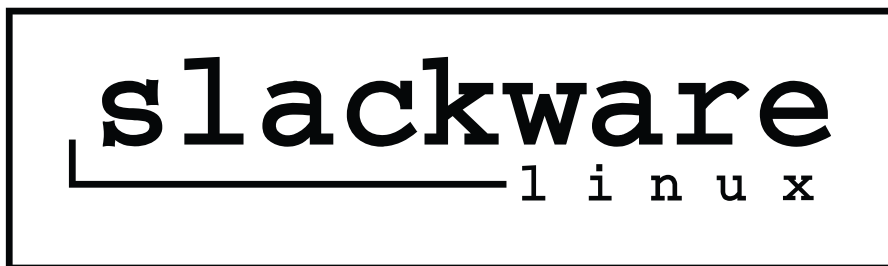


Figura 2: Até hoje o Slackware é "trabalho de um homem só", Patrick Volkerding.

adiante, que era uma distribuição para machos. Estou aqui para demonstrar que essa segunda premissa não é verdadeira. O Slackware simplesmente assume que o administrador tem o controle do sistema e que sabe o que está fazendo. Como disse Piter Punk em uma de suas palestras, ele "resgata o prazer da leitura". O sistema é indicado para usuários novatos ou experientes, desde que tenham, porém, vontade de entender o funcionamento do sistema e desejem total controle sobre o que está sendo instalado e configurado em sua máquina. Tudo isso com extrema simplicidade.

Conhecido por sua estabilidade, um dos grandes segredos do Slackware se deve ao fato de ser a única distribuição que não aplica *patches* nem adaptações aos softwares que distribui. E também de sempre lançar versões com pacotes atualizados e manter um rápido sistema de atualização, caso surjam problemas de segurança.

A instalação é toda em modo texto e os pacotes são divididos em séries: *A* (instalação base), *AP* (aplicações básicas de sistema), *D* (desenvolvimento), *K* (fontes do kernel), *N* (rede), *KDE*, *GNOME*, *L* (bibliotecas), *X*, *XAP* (aplicações gráficas). Depois de escolher as séries que deseja instalar, você seleciona o nível de controle que quer ter sobre os pacotes instalados: modo *expert* (todos os pacotes podem ser selecionados ou desativados), *menu* (pacotes essenciais ao sistema não são mostrados), *full* (instalação completa) ou por *tagfiles*. A instalação pode ser feita por CD ou através da rede, via FTP ou NFS.

O desenvolvimento do Slackware é feito exclusivamente por Volkerding. Ele toma as decisões do que entra ou não na distribuição. O que implica, por exemplo, no fato da distro não utilizar

o PAM. Volkerding tem uma profunda aversão ao PAM e prefere deixar tudo simplificado pelo NSS. O ciclo de desenvolvimento é de aproximadamente 6 meses entre as versões. A filosofia da distribuição é sempre estar concentrada na simplicidade e estabilidade. A versão atual, 10.0, trouxe algumas novidades, como o suporte ao *UDEV* (que irá substituir o *devfs*), sistema */sys* (que irá substituir o */proc*) e atualizações gerais. O kernel 2.6 continua no diretório *testing*, não tendo sido oficialmente incorporado à distribuição.

Os scripts de inicialização do Slackware seguem o padrão BSD; todos ficam no diretório */etc/rc.d/*. Os scripts principais, que carregam o sistema de acordo com o run-level escolhido, verificam as permissões de execução dos scripts de serviço (como *rc.httpd*, *rc.sshd*) e os executam, se permitido. Para desativar a execução de um serviço no boot, basta remover a permissão de execução do script correspondente.

O gerenciamento de pacotes é feito pelo *pkgtool* e seus agregados: *installpkg*, *removepkg*, *upgradepkg*, *makepkg* e *explodepkg*. Essas ferramentas já foram melhor detalhadas nas edições 1 e 2 da Linux Magazine Brasil. Resumidamente, servem para instalar, remover, atualizar ou construir pacotes do Slackware. Esses pacotes têm um formato bastante simples: são arquivos *tar.gz* contendo os binários, bibliotecas e demais arquivos que compõem o software, mais um script para executar algum comando necessário após a instalação e a descrição do software. Uma listagem desses arquivos, junto com a descrição, é colocada no diretório */var/log/packages*, num arquivo que leva o nome do pacote. Isso serve para consultas rápidas sobre versões e arquivos instalados.

O Slackware, por definição, não verifica as dependências entre pacotes.

Alguns programas, entretanto, fazem essa verificação por conta e risco do autor, como o *swaret* e o *slapt*. Outro software, o *slackpkg* [5], apenas consulta a listagem de pacotes de um mirror escolhido, verifica versões, instala ou atualiza pacotes, de acordo com o comando escolhido.

Os pacotes são feitos com base nos *SlackBuilds*, scripts que compilam o software e formatam um pacote de acordo com o padrão oficial. Eles estão sempre disponíveis junto aos fontes do Slackware. Com eles, é fácil modificar alguma opção de compilação faltante ou excedente, como por exemplo suporte a algum recurso que não venha por padrão em um pacote.

O Slackware também tem uma versão super reduzida, chamada *ZipSlack*, que pode ser instalada em qualquer partição FAT (ou FAT32) com 100Mb de espaço livre e utiliza o sistema de arquivos UMSDOS. Isto significa que você não precisa reparticionar seu HD e instalar tudo para testar o Linux e conhecer a cara do Slackware.

Com tudo isso, espero ter fornecido uma visão geral das distribuições e que alguma tenha lhe chamado a atenção. Ao download! ■

INFORMAÇÕES

- [1] Gentoo: <http://www.gentoo.org>
- [2] Gentoo-BR: <http://www.gentoobr.org>
- [3] Slackware: <http://www.slackware.org>
- [4] GUS-BR: <http://gus-br.linuxmag.com.br/>
- [5] Slackpkg: <http://slackpkg.sourceforge.net/>
- [6] Swaret: <http://swaret.sourceforge.net>
- [7] Slapt-get: <http://software.jaos.org>
- [8] Trabalhando com o Portage: http://www.gentoo.org/doc/pt_br/handbook/handbook-x86.xml?part=3&chap=0
- [9] Gentoo Handbook (em Português): http://www.gentoo.org/doc/pt_br/handbook/index.xml

SOBRE A AUTORA

Sulamita Garcia é formada em Ciências da Computação pela UFSC, onde conheceu o Linux. Participa do projeto LinuxChix e mantém a seção sobre Alta Disponibilidade do site UnderLinux. Possui certificação LPIC II e trabalha como Software Designer na Cyclades.

