Montagem: Luciano Hagge

Pacotes no Slackware

Brincando com o *slackpkg*

Inicialmente criado apenas para instalar e manter patches de segurança, com o tempo o *slackpkg* evoluiu ao ponto de poder ser usado para manter em dia todo um sistema Slackware. Mas... o que exatamente o *slackpkg* faz?.

POR PITER PUNK

A principal função do *slackpkg* é verificar se existe versão mais recente de um ou mais dos pacotes instalados em seu sistema, baixar o novo pacote e fazer a atualização para você. Com isso ganhamos vários recursos como "brinde"; afinal, se já temos uma lista de pacotes, é possível usá-la para consultar, baixar e instalar os que você ainda não tem e, eventualmente, reinstalar aqueles que já estão no sistema (vai saber o que você fez com eles para precisar reinstalar...).

Atualizar, instalar e reinstalar, é só isso que dá pra fazer? Não! Assim como nos produtos vendidos através de comerciais de TV, se ligar agora você leva muito mais! Com o *slackpkg* você pode procurar por um determinado arquivo e saber a qual pacote ele pertence ou, se preferir, consultar a descrição de um pacote antes de instalá-lo.

Obtendo o *slackpkg*

O primeiro passo é conseguir sua própria cópia, zero KM, do *slackpkg*. É fácil, basta visitar o site oficial [1] e baixar o programa, que tem cerca de 30 KB. Depois, instale-o com o utilitário *installpkg*, parte do sistema de gerenciamento de pacotes do Slackware:

```
installpkg slackpkg-versão-noarch-10.tgz
```

Pra quem não tem acesso à Internet, o programa também pode ser encontrado no diretório `extra` dos CDs do Slackware 9.1 (ou de versões mais recentes) e em todos os *mirrors* do servidor FTP oficial da distribuição. Nesse caso instalação também é feita via *installpkg*, da mesma forma citada anteriormente.

Usando o *slackpkg*

A primeira coisa a fazer é editar o arquivo `/etc/slackpkg/mirrors` e escolher um *mirror* de onde baixar os pacotes. O arquivo já contém uma lista enorme de vários servidores com pacotes para pelo menos três versões do Slackware. Para escolher um deles, basta descomentar (apagando o caractere `#`) a linha correspondente. Feito isso, vamos atualizar a lista de pacotes:

```
# slackpkg update
```

Isso vai baixar a lista com os pacotes que compõem o Slackware, suas descrições e uma terceira lista com seu conteúdo. Na primeira vez que for executada a operação *update*, o *slackpkg* também vai baixar a chave PGP do projeto Slackware, que é utilizada para verificar a autenticidade dos arquivos. Se por algum motivo você precisar atualizar ou reinstalar a chave, basta digitar:

```
# slackpkg update gpg
```

Agora que tudo está pronto, vamos brincar um pouco. Digite:

```
# slackpkg upgrade slackware
```

O comando acima vai atualizar toda a distribuição. Muito cuidado, pois o conceito de "atualização" do *slackpkg* significa: "manter os pacotes locais iguais aos do servidor espelho selecionado". Se você estiver usando o Slackware 10.1 e escolher um mirror da versão 10.0, na verdade todos os seus pacotes irão **regredir** para seus equivalentes na versão 10.0. Parabéns, você acabou de fazer um *downgrade* do sistema inteiro! Por incrível que pareça, isso pode ser útil em alguns casos.

Se você estiver usando a versão estável do Slackware e quiser baixar apenas os *patches* e atualizações de segurança, digite o comando:

```
# slackpkg upgrade patches
```

E se você tiver apenas o primeiro CD do Slackware e precisar instalar algo que está no segundo disco? Sem problemas, é possível, por exemplo, instalar todo o GNOME com o comando:

```
# slackpkg install gnome/
```

Também é possível instalar apenas o pacote de internacionalização do KDE para o português do Brasil (vulgo *pt_BR* no mundo Unix):

```
# slackpkg install kde-i18n-pt_BR
```

Brincando, você viu que o *slackpkg* consegue tratar tanto da distribuição inteira como de uma das séries de pacotes e até mesmo instalar pacotes um a um pelo nome. Isso acontece porque ele não trabalha com o nome do pacote em si, mas sim com um sistema de reconhecimento de padrões. Veja como os arquivos são organizados num *mirror* da distribuição:

```
raiz/slackware/ap/kbd-xyz.tgz
```

Assim, um `upgrade slackware` atualizará tudo que estiver debaixo do diretório `slackware` na árvore. Se trocarmos `slackware` por `ap`, serão atualizados apenas os pacotes dentro do diretório `ap` e, se

especificarmos o nome do pacote, apenas ele será instalado. Claro que isso também vale para operações como `install` ou `reinstall`.

O *slackpkg* é inteligente. Se digitar:

```
# slackpkg install ap/
```

e algum dos pacotes da série `ap` tiver uma atualização de segurança no diretório `patches`, é a ela que será instalada (a não ser que você altere a ordem de busca no arquivo de configuração do *slackpkg*, `/etc/slackpkg/slackpkg.conf`).

E não é só isso!

Algumas partes do Slackware são comumente personalizadas (como arquivos de configuração e scripts de inicialização) e não podem ser reescritas durante um *upgrade*. Isso é garantido pela própria estrutura de um pacote Slackware, onde esse tipo de arquivo tem o sufixo `.new`.

Quando um arquivo de configuração não existe (ou seja, quando o pacote é instalado pela primeira vez), o que veio com o pacote é renomeado, perde o sufixo `.new` e tudo fica bem. Quando o arquivo já existe, a versão anterior é mantida e a nova fica ao lado, como referência.

Como muita gente não costuma verificar os `.new` e movê-los para seus devidos lugares, no final das contas o sistema fica infestado de `.new` e pode deixar de funcionar direito (isso é comum com os scripts de inicialização no diretório `/etc/rc.d`).

Se você costuma se esquecer dos `.new` não se preocupe, pois o *slackpkg* não se esquece! Após instalar, atualizar ou reinstalar qualquer pacote, todo o diretório `/etc` é verificado em busca de arquivos `.new` e o usuário tem a opção de manter os arquivos de configuração como estão, colocar os `.new` em ação e preservar seus originais com o sufixo `.orig`, remover todos os `.new` ou, se preferir, dizer caso a caso o que fazer com os benditos arquivos de configuração. Se você não quer ser importunado com esse detalhe toda vez que for atualizar algum pacote, basta editar o arquivo `/etc/slackpkg/slackpkg.conf` e desativar a opção `POSTINST`.

E ainda tem mais!

Existe ainda a função de procura de arquivos, muito útil para descobrir a qual pacote pertence um arquivo. Quer saber de onde veio o *ls*?:

```
# slackpkg search ls
The list below shows all packages with
the selected pattern.
[ installed ] - bin-9.2.0-i486-2
[ installed ] - coreutils-5.2.1-i486-1
[ installed ] - dcron-2.3.3-i386-4
...
[uninstalled] - libxml-1.8.17-i486-3
[uninstalled] - modutils-2.4.25-i486-1
```

Xiii... apareceram vários arquivos (exatos 286 pacotes no Slackware 10.1); qual deles é o correto? fácil: sabemos que o *ls* é um comando, portanto digitamos:

```
# slackpkg search bin/ls
The list below shows all packages with
the selected pattern.
[ installed ] - coreutils-5.2.1-i486-1
[ installed ] - e2fsprogs-1.35-i486-1
...
[ installed ] - kdegames-3.3.2-i486-1
[uninstalled] - wu-ftpd-2.6.2-i486-3
```

Agora obtivemos apenas 12 pacotes e podemos verificar cada um deles para saber qual contém o que procuramos:

```
# slackpkg info nome_do_pacote
```

Da mesma maneira como encontramos o *ls*, podemos encontrar vários outros arquivos, como bibliotecas misteriosas:

```
# slackpkg search lib/libcaca
```

E, com isso, instalar "aquela" biblioteca que está faltando para compilar o programa *xyz*. Depois de achar o que procura e confirmar seu palpite com a operação `info`, é só usar o *slackpkg* e instalar o pacote que deseja.

Um último truque antes de darmos adeus à função de procura do *slackpkg*: se você souber exatamente o nome do arquivo que procura, pode adicionar um `[^a-z]` ou `[^[:alpha:]]` para excluir qualquer coisa após o termo de busca. Por exemplo, uma busca por ls retornaria tanto *ls* quanto *lsattr*. Não se você incluir `[^a-z]` ao final do termo. Um exemplo:

```
root@rachael:/home/punk# slackpkg search ➥
bin/ls[^[:alpha:]]
The list below shows all packages with
the selected pattern.
[ installed ] - coreutils-5.2.1-i486-1
[uninstalled] - wu-ftpd-2.6.2-i486-3
```

Alguns cuidados

Vimos vários dos recursos do *slackpkg*. Ele baixa, instala, reinstala, atualiza, pesquisa e mostra informações sobre os pacotes. Mas ele não é realmente inteligente e, às vezes, pode não fazer exatamente o que seria o melhor para sua máquina.

Por isso, todas as opções que alteram o sistema solicitam confirmação; não há risco de você atualizar um pacote por engano ou acabar instalando todo o GNOME quando queria apenas o *XChat*. No *slackpkg* você tem exatamente o que pede, nem um pacote a mais ou a menos. Controle total ou seu dinheiro de volta.

Às vezes você quer baixar os pacotes, mas não tem tempo para examiná-los com cuidado. Para facilitar esse controle, com o *slackpkg* você pode baixá-los agora:

```
# slackpkg download gnome/
```

E instalá-los mais tarde:

```
# slackpkg upgrade gnome/
```

Ainda pensando em controle, nada pior do que ter uma nova versão do kernel instalada por cima da que foi cuidadosamente compilada por você mesmo. Para evitar isso, os pacotes do kernel podem ser colocados numa lista negra, ou *blacklist*:

```
# slackpkg blacklist kernel
```

Os pacotes na *blacklist* são total e solenemente ignorados pelo *slackpkg*. Se quiser retirar algo da *blacklist*, basta editar o arquivo `/etc/slackpkg/blacklist`. Uma dica é colocar o pacote *alsa-driver* na *blacklist*, para não correr o risco do som deixar de funcionar após um upgrade.

O *slackpkg* é uma ferramenta versátil, poderosa e simples, que segue a tradição KISS (*Keep It Simple, Stupid!*) do Slackware. Enquanto outras ferramentas tentam empurrar recursos alienígenas à distribuição (como resolução de dependências), o *slackpkg* utiliza apenas as informações já disponíveis nos próprios pacotes. Além disso, baixa e instala apenas os pacotes oficiais, que têm garantia de confiabilidade e estabilidade. ■

Informações

[1] Página oficial do slackpkg: http://slackpkg.sf.net/