Gerenciamento de instalações e metapacotes com o simplepkg

Silvio Rhatto

22 de agosto de 2007

Resumo

O *simplepkg* é um sistema de gerenciamento de sistemas slackware que roda sobre o pkgtool. Ele é composto por vários scripts que desempenham funções de administração e desenvolvimento de sistemas do tipo Slackware, procurando fechar um circuito de produção, envolvendo a construção, a instalação de pacotes e até mesmo a instalação do sistema de forma automatizada.

Do lado do desenvolvedor/a, ele ajuda na criação de SlackBuilds e construção de pacotes. Do lado do administrador/a, ele possibilita a instalação automatizada de sistemas, instalação de pacotes e a criação de "templates"de instalação – que contém todos os arquivos de configuração, informações de permissões e scripts de pós-instalação de uma dada máquina ou jaula.

1 Descrição

Todas as distribuições de GNU/Linux já tem algum sistema de empacotamento amadurecido. A questão agora é a praticidade de instalar e controlar o que está instalado, tanto pacotes como arquivos de configuração de uma máquina, além da facilidade na criação de pacotes.

Imagine por exemplo se você precisa manter uma lista de pacotes de 200 máquinas slackware, sendo que algumas são usadas como desktop, outras como servidores web, alguma sendo o servidor de email e assim por diante. Imagine agora que você perca o disco de algumas dessas máquinas ou que precise cotidianamente reinstalar ou atualizar um sistema.

Usar o cd de instalação do slackware e configurar na mão toda a vez que der um pau faria com que você ficasse louco/a e desperdiçasse muito tempo, além do que sempre ocorre de esquecermos algum detalhe ou pacote durante a configuração do sistema. Manter um backup completo de cada máquina, por outro lado, pode ser muito custoso se o número delas for muito grande.

O *simplepkg* permite que você mantenha um template para cada grupo de máquinas e com apenas um comando instalar o template numa partição. Além

do template, você precisa configurar o *simplepkg* para obter pacotes de um repositório local ou remoto.

Gerenciar instalações e pacotes não é tudo o que o *simplepkg* faz. Ele pode ser usado até na criação de jaula e vservers, podendo manter toda a configuração das máquinas num repositório Subversion.

O *simplepkg* funciona não apenas com o Slackware mas com qualquer port (oficial ou não) que siga minimamente os padrões oficiais.

2 Arquitetura

O *simplepkg* é um conjunto de scripts escritos com a filosofia KISS (Keep It Simple, Stupid) em mente. Ele é um sistema muito simples, composto pelos seguintes comandos:

- mkjail: constrói uma jaula/instalação de slackware numa pasta
- templatepkg: criar ou adiciona pacotes a um template
- lspkg: lista pacotes instalados
- jail-update: inverso do jail-commit
- jail-commit: atualiza o template
- rebuildpkg: reconstrói um pacote a partir de sua entrada no /var/log/packages
- simplaret: obtém pacotes de repositórios locais ou remotos
- createpkg: baixa, compila e empacota software de acordo com scripts presentes num repositório
- repos: cria e mantém repositórios
- mkbuild: cria scripts de construção de pacotes

3 Instalando o simplepkg

Para baixar o pacote do *simplepkg*, vá em http://slack.sarava.org/packages/noarch/. Depois, basta usar o comando

```
installpkg simplepkg-VERSAO-noarch-BUILD.tgz
```

4 Usando o simplepkg

As três principais aplicações desse conjunto são:

- Gerenciamento de pacotes

- Criação e manutenção de jaulas
- Criação de pacotes

O gerencamento de pacotes é feito através do simplaret, e por ser algo bem específico está detalhado no artigo correspondente. As seções a seguir mostrarão como o *simplepkg* pode ser utilizado para criar e manter jaulas, assim como também criar pacotes.

5 Criando templates de instalação

Originalmente, o *simplepkg* foi desenvolvido para ajudar na automatização de instalações de sistemas slackware. Para isso, ele trabalha com templates – listas com pacotes instalados, scripts e arquivos de configuração – permitindo criar perfis de instalação que podem ser então usados para instalar o sistema numa outra partição ou criar um chroot.

A construção de um template é feita através do programa templatepkg. Para criar um template de nome "meu-slackware"contendo a lista de pacotes atualmente instalados no seu sistema, digite

```
templatepkg -c meu-slackware
```

A opção -c (ou –create) criará a pasta /etc/simplepkg/templates/meu-slackware, que conterá os seguintes componentes:

- */etc/simplepkg/templates/meu-slackware/meu-slackware.d*: cópia de arquivos de configuração
- */etc/simplepkg/templates/meu-slackware/meu-slackware.s*: scripts de pós-instalação
- */etc/simplepkg/templates/meu-slackware/meu-slackware.perms*: informações sobre arquivos
- */etc/simplepkg/templates/meu-slackware/meu-slackware.template*: lista de pacotes

Esses quatro componentes são suficientes para armazenar todas as características de uma instalação de slackware: a lista de pacotes controla o software instalado (a partir do conteúdo da pasta */var/log/packages*), a cópia dos arquivos de configuração controla as personalizações feitas para o uso dos aplicativos e os scripts de pós-instalação cuidam de qualquer rotina que precisa ser realizada exatamente após a instalação do sistema. Já o arquivo de informações sobre arquivos contém as permissões, o dono/a e grupo de cada arquivo de configuração presente no template.

Se você quiser criar um template a partir de uma instalação de slackware presente numa outra partição do sistema que não seja a raíz, basta usar um comando do tipo

```
templatepkg -c meu-slackware /mnt/slackware
```

onde /mnt/slackware é o local onde o sistema alternativo está instalado. Após criado, o template possuirá apenas a lista de pacotes contendo o nome dos aplicativos instalados no seu sistema. Como a pasta /var/log/packages não preserva a ordem de instalação dos pacotes, então talvez você queira editar manualmente a lista de pacotes de um template. Para isso, use o comando

```
templatepkg -e meu-slackware
```

Para adicionar um arquivo de configuração no seu novo template, basta dar um comando como

```
templatepkg -a meu-slackware /etc/hosts
```

Isso adicionará o arquivo /etc/hosts no template "meu-slackware". Além de salvar o arquivo e copiá-lo automaticamente quando você instalar seu sistema, o *simplepkg* ainda pode tomar conta de qualquer alteração que o /etc/hosts sofrer no sistema, seja mudança no conteúdo do arquivo, dono ou permissão. Se você ainda estiver armazenando seus templates num repositório svn (o que veremos a seguir), o *simplepkg* pode ainda manter um histórico completo das alterações do arquivo.

ATENÇÂO: evite ao máximo deixar arquivos contendo senhas ou chaves privadas num template. O lugar mais adequado para deixar esse tipo de coisa é num backup seguro.

6 Criando jaulas e replicando instalações

Uma vez que um template foi criado com uma lista de pacotes e opcionalmente com arquivos de configuração e scripts de pós-instalação (que serão detalhados a seguir), você pode replicar sua instalação de slackware utilizando o comando

```
mkjail jaula meu-slackware
```

Isso cria uma nova árvore do slackware em /vservers/jaula contendo todos os pacotes e arquivos de configuração do template "meu-slackware". A instalação dos pacotes será feita pelo aplicativo simplaret, que deve estar configurado corretamente e cuja configuração padrão deve funcionar para a maioria dos casos.

Se você quiser instalar essa jaula em outro local que não seja a pasta /vservers (esse local padrão pode ser mudado pelo arquivo de configuração do *simplepkg*), basta usar um comando do tipo

```
ROOT=/mnt mkjail hda2 meu-slackware
```

O comando acima faz exatamente o que você está pensando: replica sua instalação slackware em /mnt/hda2, dispensando totalmente o programa de instalação do slackware!

Caso nenhum template for especificado, o mkjail utiliza o template /etc/simplepkg/default.
O *simplepkg* já vem com alguns templates padrões, presentes em /etc/simplepkg/defaults/templates.

7 Scripts de pós-instalação

Opcionalmente, é possível manter scripts de pós-instalação num template. Tais script são executados exatamente após a instalação de uma jaula e cópia de arquivos de configuração pelo mkjail. Para criar ou editar um script de pós-instalação, use um comando como

```
templatepkg -b meu-slackware nome-do-script.sh
```

Isso adicionará o script nome-do-script.sh no template "meu-slackware". O mkjail passa dois argumentos para cada script de pós-instalação: a pasta superior e o nome da jaula ("/mnt"e "hda2"no nosso exemplo anterior). Assim, um exemplo de script seria algo como

```
#!/bin/bash
chroot $1/$2/ sbin/ldconfig
```

8 Listando o conteúdo de um template

Para listar os templates disponíveis ou o conteúdo de um template, use comandos como

```
templatepkg -l
templatepkg -l meu-slackware
```

9 Removendo arquivos de um template

Analogamente à forma como se adicona arquivos num template, removê-los pode ser feito com o seguinte comando:

```
templatepkg -d meu-slackware /etc/hosts
```

Isso remove o arquivo /etc/hosts do template "meu-slackware".

10 Apagando um template

Para apagar um template, basta utilizar um comando como

```
templatepkg -r meu-slackware
```

11 Atualizando um template

Agora que já abordamos as opções do templatepkg, é hora de visitarmos um outro aplicativo, desta vez utilizado para manter um template atualizado. O jail-commit é um script que copia as alterações dos arquivos (conteúdo, propriedade e permissões) de um template a partir do conteúdo de uma jaula ou instalação.

Por exemplo, caso se queira copiar as alterações da jaula /mnt/hda2 no template "meu-slackware", basta usar o comando

```
jail-commit /mnt/hda2 meu-slackware
```

Além da lista de pacotes do template "meu-slackware"ser atualizada de acordo com a lista de pacotes presente em /mnt/hda2/var/log/packages, todos os arquivos de configuração presentes no template "meu-slackware"serão comparados com os correspondentes da pasta /mnt/hda2 e as diferenças são copiadas para o template. Da mesma forma, as permissões e informação de dono/grupo dos arquivos também é atualizada no template.

O comando jail-commit possibilita que um template sempre esteja atualizado e refletindo a configuração atual de uma instalação de slackware. Se você quiser atualizar apenas a lista de pacotes de um template, porém, use

```
templatepkg -u meu-template
```

Para facilitar ainda mais o controle das alterações do sistema, existe ainda uma facilidade do arquivo /etc/simplepkg/jailist. Esse arquivo serve, além de outros propósitos descritos na documentação do simplaret, para que o jail-commit saiba de antemão quais são as instalações de sistema do tipo Slackware presentes numa máquina, além da instalação principal na raíz do sistema.

Suponha que uma máquina possua duas instalações de slackware, além da principal (raíz):

- /mnt/slackware-1 usando o template "slackware-1"
- /mnt/slackware-2 usando o template "slackware-2"

Se o arquivo /etc/simplepkg/jailist contiver as seguintes linhas,

```
/mnt/slackware-1
/mnt/slackware-2
```

então o comando

```
jail-commit
```

atualizará o template "slackware-1"de acordo com o conteúdo da jaula /mnt/slackware-1 e o template "slackware-2"com o conteúdo da jaula /mnt/slackware-2. Se, além desses dois templates, existir um outro de nome "main", então o jail-commit sem argumentos também copiará as atualizações da instalação raíz, deixando-as no template "main".

Você pode inclusive colocar uma entrada no crontab do tipo

```
20 4 * * * jail-commit
```

para que seus templates sejam atualizados diariamente. Se você ainda possui o envio de emails configurado na sua máquina, então a saída do jail-commit será enviada pelo cron diariamente para seu email, contendo diffs das alterações de arquivos de configuração a lista de pacotes adicionados ou removidos no sistema.

12 Restaurando arquivos de configuração numa jaula

A operação contrária ao que o jail-commit faz também é possível: suponha que você mexeu na configuração do sistema mas se arrependeu das alterações e deseja voltar a configuração para o modo como ela se encontra no seu template, basta usar o comando

```
jail-update /mnt/hda2 meu-slackware
```

13 Armazenando as configurações no repositório Subversion

Para aumentar ainda mais a flexibilidade e o controle do conteúdo dos templates, é possível armazená-los num repositório Subversion. Para isso, edite o arquivo /etc/simplepkg/simplepkg.conf e deixe o parâmetro de configuração *TEMPLATES_UNDER_SVN* com o valor "yes".

Depois, crie um repositório subversion para armazenar seus templates, usando algo como

```
svnadmin create /var/svn/simplepkg --fs-type fsfs
```

Com o repositório criado, basta importar seus templates com o comando

```
templatepkg -e file:///var/svn/simplepkg
```

A partir daí, o comando jail-commit enviará automaticamente todas as alterações dos templates para o repositório subversion. Se, por outro lado, você quiser baixar as alterações dos templates que estão no repositório remoto para sua cópia local, use o comando

```
templatepkg -s
```

Caso você queira importar uma pasta de templates de um repositório já existente, use

```
templatepkg -i file:///var/svn/simplepkg
```

onde file:///var/svn/simplepkg é o camninho do repositório.

14 Atualização de jaulas

A atualização de jaulas e sistemas instalados é feita através do simplaret e também utiliza o arquivo /etc/simplepkg/jailist. Para mais informações a respeito, consulte a documentação do simplaret para mais detalhes.

15 Arquiteturas e versões diferentes

O *simplepkg* foi idealizado para permitir que um mesmo template possa ser usado para criar jaulas de arquiteturas e versões diferentes de sistemas padrão slackware. A atualização desses sistemas também é unificada. Essa possibilidade só é permitida se você usa o simplaret e não o swaret como ferramenta de obtenção de pacotes.

Por exemplo, para criar uma instalação de slackware 10.1 (assumindo que suas definições de repositórios do /etc/simplepkg/repos.conf contenham locais com slack 10.1), basta usar o comando

```
VERSION=10.1 mkjail minha-jaula template-servidor
```

Arquiteturas diferentes também podem ser usadas. Se você está num sistema x86_64 e quer instalar um slack 10.2 numa partição, experimente

```
ARCH=i386 VERSION=10.2 ROOT=/mnt mkjail hda2 meu-slackware
```

Note que os templates independem de arquitetura e versão, já que eles só contém nomes de pacotes, arquivos de configuração e scripts.

16 Criando um pacote de um template

Se, por algum motivo, você quiser construir um pacote com o conteúdo de um template, experimente o comando

```
templatepkg -p nome-do-template
```

No entanto, o *simplepkg* foi criado para que esse tipo de comportamento seja evitado, já que é mais simples manter templates de configuração do que pacotes contendo a configuração de uma instalação.

17 Construindo pacotes

Até aqui, apenas mostramos os aplicativos do *simplepkg* usados para a manutenção de instalações de slackware. No entanto, uma das outras finalidades desta suíte é a construção de pacotes, o que é feita pelo programa createpkg, Como dito anteriormente, o createpkg: baixa, compila e empacota software de acordo com scripts presentes num repositório de scripts, funcionando com um gerenciador de "ports"para slackware.

O createpkg pode funcionar com qualquer tipo de script de construção de pacotes (SlackBuilds) mas funcionará melhor se os mesmos seguirem o padrão descrito na página

```
http://slack.sarava.org/trac/wiki/SlackBuilds
```

Especificamente, o createpkg foi desenvolvido para utilizar os slackbuild disponíveis em http://slack.sarava.org/slackbuilds. O createpkg trabalha com repositórios do tipo subversion.

Para obter os scripts do repositório do slack.sarava.org, digite

```
createpkg --sync
```

Em seguida, você pode listas todos os scripts disponíveis:

```
createpkg --list
```

Para buscar por um pacote, use

```
createpkg --search latex2html
```

No caso, a busca é feita pelo SlackBuild do aplicativo "latex2html". Suponha agora que você queira construir o pacote desse aplicativo:

```
createpkg latex2html
```

O pacote resultante estará na pasta /tmp ou no valor especificado pela variável de ambiente *$REPOS*. Para criar e também instalar o pacote, basta

```
createpkg --install latex2html
```

Se o pacote possuir dependências listadas num arquivo slack-required e que não estiverem instaladas no sistema, o createpkg tentará processá-las uma a uma antes de tentar construir o pacote desejado: se as dependências não forem encontradas no repositório de scripts, então o createpkg tentará baixá-las de um repositório binário através do simplaret. Se você não quiser que a resolução de dependências seja seguida, use a opção –no-deps.

Para mais detalhes de funcionamento, experimente o comando

```
createpkg --help
```

ou então acesse a página http://slack.sarava.org/trac/wiki/SlackBuilds

18 Aplicativos auxiliares

O *simplepkg* acompanha ainda alguns aplicativos auxiliares:

- lspkg: lista pacotes instalados
- rebuildpkg: reconstrói um pacote a partir de sua entrada no /var/log/packages
- repos: cria e mantém repositórios
- mkbuild: cria scripts de construção de pacotes

O comando lspkg é um utilitário simples para a visualização de pacotes instalados no sistema. Já o rebuildpkg ajuda a recuperar pacotes instalados cujo tgz original foi perdido. O comando rebuildpkg reconstrói um pacote a partir de uma entrada no /var/log/packages. O comando

```
rebuildpkg coreutils
```

reconstrói um pacote do coreutils usando os arquivos e as metainformações listadas no arquivo do /var/log/packages/ correspondente ao coreutils.

Por fim, os scripts repos e mkbuild são os que se encontram na etapa de maior desenvolvimento: repos cria um repositório de pacotes a partir de uma pasta contendo pacotes do tipo pkgtool e o mkbuild é um aplicativo para auxiliar a criação de scripts de construção de pacotes que podem ser utilizados sozinhos ou pelo o createpkg.

19 Parâmetros de configuração

O arquivo de configuração do *simplepkg* é o /etc/simplepkg/simplepkg.conf. Ele contém parâmetros de configuração de todos os scripts, porém neste texto não trataremos das opções específicas ao simplaret, as quais tem uma seção específica no artigo correspondente.

- *JAIL_ROOT*: pasta padrão onde as jaulas são criadas pelo mkjail. Valor padrão: "/vservers".
- *ADD_TO_JAIL_LIST*: controla se uma jaula criada pelo mkjail deve ser adicionada automaticamente no arquivo /etc/simplepkg/jailist. O valor padrão é "1"(habilitado).
- *TEMPLATES_UNDER_SVN*: indica se os templates estão armazenados num repositório subversion. O valor padrão é "no"(não).
- *TEMPLATE_FOLDER*: indica qual é a pasta de templates. O valor padrão é "/etc/simplepkg/templates"e não é recomendável alterá-lo.

- *TEMPLATE_STORAGE_STYLE*: controla a forma de armazenamento de templates. O valor padrão é "own-folder"e essa opção apenas deve ser modificada se você armazena seus templates num formato antigo do *simplepkg* e deseja manter compatibilidade.

Vale lembrar que todas as opções booleanas (isto é, que podem ser apenas habilitadas ou desabilitadas) do simplepkg.conf tem os seguintes valores permitidos: "1"e "yes"para habilitado e "0"ou "no"para desabilitado.

20 Mais informações

O *simplepkg* foi escrito por Silvio Rhatto (rhatto at riseup.net) sob licença GPL e seu código fonte é disponibilizado através do repositório subversion:

```
svn checkout svn://slack.sarava.org/simplepkg
```

O wiki de desenvolvimento está em http://slack.sarava.org/trac/wiki/Simplepkg e o endereço da lista de discussão utilizada para discussões sobre o *simplepkg* ou mesmo distribuições e pacotes do tipo Slackware é http://listas.sarava.org/wws/info/slack.