

# Gerenciamento de instalações e metapacotes com o simplepkg

Silvio Rhatto

17 de janeiro de 2006

## Resumo

O *simplepkg* é um sistema de gerenciamento de pacotes que roda sobre o *pkgtool* e o *swaret*. 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, criar um chroot específico ou até trabalhar com o conceito de *metapacotes*: uma lista de pacotes que pode ser instalada e removida com apenas um comando.

## 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.

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 do *swaret* configurado para os repositórios de pacote que possuem seus aplicativos.

Gerenciar instalações e metapacotes não é tudo o que o *simplepkg* faz. Ele pode ser usado até na criação de *servers*. O *simplepkg* inclusive é um amadurecimento de idéias delineadas nos seguintes artigos:

- Rodando o OpenOffice.org em chroot no Slamd64

- Criando Vservers em Slackware
- Instalando o Slackware sem programa de instalação

O *simplepkg* permite que o trampo proposto nos artigos acima seja simplificado a uma meia dúzia de comandos, desde que existam templates prontos.

## 2 Arquitetura

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

- `mkjail`: constrói uma jaula/instalação de slackware numa pasta
- `metapkg`: instala ou remove um metapacote
- `templatepkg`: criar ou adiciona pacotes a um template
- `lspkg`: lista pacotes instalados
- `jail-upgrade`: faz o upgrade nas jaulas
- `rebuildpkg`: reconstrói um pacote a partir de sua entrada no `/var/log/packages`

A pasta de configuração e armazenamento de templates é a `/etc/simplepkg`. Templates são arquivos contendo uma lista de pacotes, um pacote por linha e com a extensão `.template`. Como exemplo, um template poderia se chamar `minimo.template` e conter a seguinte lista:

```
aaa_base
aaa_elflibs
apache
bash
bin
coreutils
findutils
```

Se você quiser também é possível usar um *tagfile* do slackware como template, sem nem precisar editá-lo.

## 3 Instalando o simplepkg

Para baixar o pacote do `simplepkg`, vá em <http://slack.midiatatica.org/packages/noarch/>. Depois de instalá-lo, não esqueça de configurar seu *swaret*:

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

Alternativamente, se você já está com o `swaret` instalado, adicione em seu `swaret.conf` a linha

```
REPOS_ROOT=SlackMidiataticaNoarch%http://slack.midiatatica.org/packages/noarch
```

e então digite

```
swaret --update
swaret --install simplepkg
```

## 4 Configurando o simplepkg

Como exemplo, vamos criar uma jaula baseada nos pacotes instalados no seu slackware. Edite o arquivo `/etc/simplepkg/simplepkg.conf`:

```
# /etc/simplepkg/simplepkg.conf
JAIL_ROOT="/vservers" # local onde as jaulas serão criadas
SWARET_PACKS="/var/swaret" # local onde o swaret armazena seus pacotes
SWARET_CLEAN="1" # apaga o cache de pacotes antes da instalação da jaula
SWARET_DELETE_DOWN="1" # apaga o cache de pacotes após a instalação
SWARET_UPDATE="0" # executa um swaret --update antes de iniciar a instalação da jaula
SWARET_PURGE_WEEKS="N" # apaga os pacotes do cache mais velhos que N semanas antes de instala
PATCHES_DIR="/var/swaret/patches" # local os patches são armazenados
```

O `swaret.conf` deve estar bem configurado para que ele seja capaz de baixar e instalar todos os pacotes dos templates. Parâmetros como `EXCLUDE` podem ser um empecilho, e se você os remover da sua instalação tenha muito cuidado ao atualizar o seu sistema via `swaret -upgrade`.

## 5 Criando jaulas e replicando instalações

```
templatepkg meu-slackware
mkjail jaula meu-slackware
```

Isso cria o template `meu-slackware` a partir da sua lista de pacotes em `/var/log/packages` e cria uma nova árvore com esses pacotes na pasta `/vservers/jaula` (dependendo de qual o valor da variável `JAIL_ROOT` do `simplepkg.conf`, é claro).

Se você quiser que alguns arquivos de configuração e executar scripts após a instalação dessa jaula, basta colocarmos na pasta `/etc/simplepkg/meu-slackware.d/` os arquivos de configuração (dentro da hierarquia de pastas do sistema, isto é, `/etc/simplepkg/meu-slackware.d/etc/apache/httpd.conf`) e os script em `/etc/simplepkg/meu-slackware.s`. O template `vserver`, que já vem no `mkjail`, contém um exemplo de script.

Você pode especificar também destinos alternativos para sua jaula, através de 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.template`.

Se você manter uma jaula em `JAIL_ROOT` com o mesmo nome que um template, é possível ainda usar o script `jail-update` para copiar todas as alterações de arquivos da jaula na pasta de arquivos do seu template (`/etc/simplepkg/nome-da-jaula.d`). Assim, basta que você copie todos os arquivos de configuração que você editou para essa pasta e deixar o `jail-update` numa crontab para que você tenha sempre um template atualizado e baseado nas jaulas / instalações que você tem rodando e que estiverem listadas no arquivo `/etc/simplepkg/jailist`:

```
jail-update
```

O `jail-update` considera que o template `main`, caso exista, se refere à instalação principal da sua máquina, isto é, o sistema de você roda. Dessa forma a invocação do `jail-update` atualiza todos os templates cujas jaulas estão em `JAIL_ROOT` e ainda atualiza o template da instalação principal.

Para adicionar ou remover pacotes de um template, basta editar o arquivo de template. Uma outra forma de adicionar arquivos é usando o `templatepkg` com a opção `-a`, que adiciona apenas novos pacotes ao template. Se você quiser que o `templatepkg` crie um template a partir de uma jaula, simplesmente use

```
templatepkg jaula /vservers/jaula
```

## 6 Metapacotes

Outro uso dos templates é a criação de pacotes: imagine por exemplo a dificuldade de um usuário/a novo de slackware para instalar o cinelerra e todas as suas dependências. Criando um template contendo o nome de todos os pacotes necessários para rodar esse software permite que um usuário instale-o simplesmente com o comando

```
metapkg --install cinelerra
```

O comando sugere uma semelhança com o uso direto do `swaret` ou do `slapt-get`, com a única diferença residindo no modo como cada um desses programas lida com as dependências de um pacote. Os `slapt-get` suporta o arquivo `slack-required`, que pode ou não estar presente no pacote. Já o `swaret` possui uma ferramenta própria para a criação de uma lista de dependências.

O `simplepkg` não pretende substituir esses dois modos de checagem e sim propor uma alternativa de distribuição de pacotes em que as dependências são resolvidas sem precisar de um repositório de dependências ou uma modificação no próprio pacote, que é o caso do `slack-required`, um arquivo que na maioria das vezes não está presente.

A remoção de um aplicativo e todas as suas dependências podem ser feitas simplesmente com o comando

```
metapkg --remove cinelerra
```

## 7 Upgrade de jaulas

O upgrade de jaulas que estejam em *JAIL\_ROOT* pode ser efetuado com o comando *jail-upgrade*. Simplesmente coloque os patches em *PATCHES\_DIR* e dê o comando

```
jail-upgrade
```

Se você quiser atualizar apenas uma das jaulas que estão em *JAIL\_ROOT*, use

```
jail-upgrade nome-da-jaula
```

Para especificar uma pasta contendo patches diferente de *PATCHES\_DIR* (útil quando você possui jaulas com diferentes versões do slackware), simplesmente use

```
PATCHES=/local/dos/patches jail-upgrade nome-da-jaula
```

De modo análogo ao *mkjail*, é possível indicar ao *jail-upgrade* uma outra pasta onde a jaula está armazenada:

```
ROOT=/otherroot jail-upgrade
```

## 8 Aplicativo auxiliar: rebuildpkg

O *simplepkg* acompanha ainda um aplicativo auxiliar que 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*.

## 9 Fonte

Se você quiser obter diretamente o código-fonte do *simplepkg*, baixe-o do repositório:

```
svn checkout svn://slack.midiatatica.org:40/simplepkg
```

## 10 Página de desenvolvimento

A página do *simplepkg* fica em <http://slack.midiatatica.org/wiki/Main/SimplePKG>.