

# GERENCIAMENTO DE PACOTES COM O APT-GET.

*Parte 01: apt-get*

*Parte 02: apt-cache*

## PARTE 01: APT-GET

Atualizar a lista de pacotes disponíveis no repositório, o apt vai pegar uma lista com todos os programas do repositório para sua máquina, por isso, para manter o cache atualizado, você deve rodar esse comando antes de uma operação com o cache ou instalar um programa.

```
apt-get update
```

---

Atualizar todos os pacotes do sistema, mas em nenhuma circunstância os pacotes presentemente instalados serão removidos, nem pacotes já instalados serão obtidos e instalados. Os pacotes presentemente instalados com novas versões e que não possam ser atualizados sem alterarem o estado da instalação de outro pacote serão deixados na versão presente.

```
apt-get upgrade
```

---

Atualizar tudo similar o upgrade, mas esse vai remover pacotes para a tarefa, ou seja, ele vai lidar com as dependências e vai atualizar a qualquer custo.

```
apt-get dist-upgrade
```

Ele comumente usado para atualizar uma versão do Debian, antes de fazer esse tipo atualização em locais críticos, leia as notas de lançamento da versão, provavelmente você irá encontrar dicas de como fazer isso de forma segura. Nós manuais antes do dist-upgrade é recomendável rodar primeiro o upgrade.

```
apt-get upgrade  
apt-get dist-upgrade
```

---

Instalar um pacote:

```
apt-get install pacote
```

Vamos a alguns exemplos:

Vou instalar o pacote tree, esse é um programa para mostrar uma lista de diretórios e arquivos em formato de árvore genealógica.

```
apt-get install tree
```

O você pode fazer uma coisa curiosa aqui, você pode remover um pacote com o install. Passando o sinal de “-” após o nome do pacote:

```
apt-get install tree-  
Lendo listas de pacotes... Pronto  
Construindo árvore de dependências  
Lendo informação de estado... Pronto  
Os pacotes a seguir serão REMOVIDOS:  
tree
```

Você também pode passar um nome de versão para o pacote a ser instalado, com o sinal de “=”.

```
apt-get install tree=1.7.0-3  
O apt-get por padrão não faz downgrade, ou seja, ele não instala uma versão velha do pacote só uma versão recente, mas se você passar uma versão menor de um pacote instalado o pacote vai regredir de versão (CUIDADO).
```

Você também pode indicar de qual versão do debian você deseja pegar o pacote, exemplos:

```
apt-get install tree/jessie-backports  
apt-get install tree/testing  
apt-get install tree/unstable
```

OBS: Se você proceder com o comando acima, talvez terá problemas com dependências, a forma mais recomendada é:

```
apt-get -t jessie-backports install pacote
```

Você também pode reinstalar o mesmo pacote, caso ele apresente algum problema.

```
apt-get install --reinstall pacote
```

---

Remover um pacote, esse comando remove o pacote, mas mantém os arquivos de configuração no sistema.

```
Apt-get remove pacote
```

Uma curiosidade é que você pode usar o remove, para instalar, passando o comando “+”.

```
apt-get remove tree+  
Lendo listas de pacotes... Pronto  
Construindo árvore de dependências  
Lendo informação de estado... Pronto  
Os NOVOS pacotes a seguir serão instalados:  
tree
```

---

Remover o pacote e também seus arquivos de configuração:

```
apt-get purge pacote
```

---

Fazer o download de um pacote fonte, você pode compilar pacotes com o apt :), para isso você vai precisar das linhas deb-src no sources.list. Se você não sabe muito sources.list, leia por favor:

<http://goo.gl/37IBZE>

Para fazer apenas o download:

```
apt-get source pacote
```

Para compilar um pacote, a melhor maneira é usar o comando **build-dep**, ele irá baixar todas as dependências necessárias para compilação, depois é só usar o comando source para compilar.

Vou compilar o pacote tree, minha cobaia geralmente nos tutoriais :)

Vou primeiro baixar todas as dependências para compilação.

```
apt-get build-dep tree
```

Agora com o comando source, eu vou baixar o arquivo fonte do pacote tree, e compilar com o comando:

```
apt-get source tree --compile
```

Isso vai gerar pacotes .deb compilados localmente no diretório corrente.

---

Ferramenta check mostra os problemas no sistema de pacotes atuais do sistema, muito útil.

```
apt-get check
```

---

Fazer o download de um pacote do repositório sem instalar, você terá no diretório corrente um pacote deb.

```
apt-get download pacote
```

---

Clean limpa o repositório local dos arquivos de pacotes obtidos remove tudo exceto o ficheiro lock de /var/cache/apt/archives/ e /var/cache/apt/archives/partial/.

```
apt-get clean
```

---

Autoclean Tal como o clean, autoclean limpa o repositório local de Arquivos de pacotes obtidos. A diferença é que apenas remove arquivos de pacotes que já não podem ser mais baixados, e são na maioria dos casos inúteis.

```
apt-get autoclean
```

---

autoremove é usado para remover pacotes que foram instalados automaticamente para satisfazer dependências de outros pacotes e que já não são necessários.

```
apt-get autoremove
```

---

Instalar programas, mas sem instalar os pacotes recomendados que é padrão no apt-get.

Isso é útil quando você precisa de um programa que faz parte de um metapacote, se você instalar pelas vias normais, vai baixar muita coisa inútil junto, consumindo espaço de disco, para resolver o problema use o parâmetro `--no-install-recommends`.

```
apt-get --no-install-recommends install pacote
```

---

O comando `-install-suggests` instala todos os pacotes sugeridos, veja PACOTES SUGERIDOS SÃO DIFERENTE DE RECOMENDADOS, pacotes sugeridos, podem ser plugins ou arquivos que expanda a funcionalidade do software, mas não são necessários para o funcionamento do programa, mas pode melhorá-lo. O apt-get nem o aptitude instala por padrão pacotes sugeridos.

```
apt-get -install-suggests install pacote
```

---

Corrigir um sistema de pacotes quebrado, esse comando é normalmente usado quando se instala um pacote .deb manualmente, e o mesmo requer dependências para o funcionamento.

Exemplo:

```
dpkg -i programa-exemplo.deb
```

O programa programa-exemplo depende das dependências A, B e C , por favor rode o comando `apt-get install -f`

```
apt-get install -f
```

Baixando e instalando dependências A, B e C....

Se o apt-get encontrar as dependências necessárias para o pacote, ele irá instalar, mas se ele não encontrar, removerá imediatamente o pacote.

---

Instala um pacote já existente no cache de pacotes, isso quer dizer que o apt vai pegar o pacote já presente no sistema e não vai realizar o download, isso é padrão no apt.

```
apt-get --no-download install tree
```

---

Simular uma instalação e remoção com o apt, você pode simular uma remoção ou instalação com apt-get, você não precisa nem logar-se como root ou usar o sudo, basta como usuário comum rodar, o comando de instalação ou remoção, seguido de `-s`.

```
apt-get install pacote -s
apt-get remove pacote -s
apt-get purge pacote -s
```

Ótimo para determinar algo antes de realmente executar.

---

Por padrão o apt-get irá perguntar se deseja instalar um programa que tenha algumas dependências para ser instalado, se o programa for simples e o sistema já ter o que ele precisa, o apt-get irá instalar automaticamente sem perguntar.

Para programas que tenham dependências, você pode usar “-y” que assume sim e não vai pergunta nada.

```
apt-get install pacote -y
apt-get remove pacote -y
apt-get purge pacote -y
```

Existem também o `--assume-no` que faz basicamente o contrário, irá perguntar sempre.

---

## ARQUIVOS :

`/etc/apt/sources.list`

Localizações de onde obter pacotes. Item de Configuração:  
Dir::Etc::SourceList.

`/etc/apt/sources.list.d/`

Arquivos fragmentados para localizações de onde obter pacotes.  
Item de Configuração: Dir::Etc::SourceParts.

`/etc/apt/apt.conf`

Arquivos de configuração do APT. Item de Configuração:  
Dir::Etc::Main.

`/etc/apt/apt.conf.d/`

Arquivos de configuração fragmentados do APT. Item de Configuração: Dir::Etc::Parts.

`/etc/apt/preferences`

Arquivos de preferências de versão. Isto é onde você deve especificar "pinning", isto é, uma preferência para obter certos pacotes A partir de uma fonte separada ou a partir duma versão diferente de uma distribuição. Item de Configuração:  
Dir::Etc::Preferences.

`/etc/apt/preferences.d/`

Arquivos fragmentados para as preferências de versão. Item de Configuração: Dir::Etc::PreferencesParts.

`/var/cache/apt/archives/`

Área de armazenamento para Arquivos de pacotes obtidos. Item de Configuração: Dir::Cache::Archives.

/var/cache/apt/archives/partial/

Área de armazenamento para Arquivos de pacotes em curso. Item de Configuração: Dir::Cache::Archives será implicitamente acrescentado (partial)

/var/lib/apt/lists/

Área de armazenamento para informação de estado para cada recurso de pacote especificado em sources.list(5) Tem de Configuração: Dir::State::Lists.

/var/lib/apt/lists/partial/

Área de armazenamento para informação de estado em trânsito. Item de Configuração: Dir::State::Lists será implicitamente acrescentado (partial)

---

## PARTE 02: APT-CACHE

O comando apt-cache, usa o cache de pacotes do apt, para procurar padrões que você passa, vamos ver na prática.

Procurar por um pacote.

**apt-cache search pacote**

---

O parâmetro showpkg, mostra várias informações sobre um pacote, como por exemplo as dependências necessárias.

**apt-cache showpkg pacote**

---

O parâmetro stats mostra o estado do cache.

**apt-cache stats**

---

O parâmetro showsrc, faz o mesmo que o showpkg, mas para pacotes fonte.

**apt-cache showsrc pacote**

---

O parâmetro depends mostra todas as dependências de um pacote

**apt-cache depends pacote**

---

O parâmetro `rdepends` mostra todas as dependências reversas, ou seja, dependências que as dependências do pacote precisam para funcionar.

```
apt-cache rdepends pacote
```

---

O parâmetro `pkgnames` mostra o nome de todos os pacotes relacionados com a busca.

```
apt-cache pkgnames pkgnames
```

---

O parâmetro `policy`, mostra a prioridade do pacote, por exemplo todos os pacotes do repositório estável tem prioridade 500, mas o `backports` tem prioridade 100, é por essa prioridade que nada no `backports` é instalado automaticamente.

```
apt-cache policy pacote
```

---

O parâmetro `madison`, mostra de que repositório o pacote vem, pegar o tree de exemplo :)

```
apt-cache madison tree
```

```
tree | 1.7.0-3 | http://httpredir.debian.org/debian/ jessie/main amd64 Packages
```

```
tree | 1.7.0-3 | http://httpredir.debian.org/debian/ jessie/main Sources
```

Aqui ele mostra que o pacote `tree` vem de um repositório de binários, `amd64` e também de repositório de fontes.

---

Autor: Fernando Debian.

Comunidade Debian Brasil no facebook, acessem:

<https://goo.gl/qfNjkj> -- <https://goo.gl/9gv0zp>