

Empacotamento de Software no Debian

Mini Aula

Formatos de números de versão de pacotes no Debian

(junho de 2020)

João Eriberto Mota Filho
eriberto@debian.org
<http://debianet.com.br>

Empacotamento de Software no Debian

Mini Aula
Formatos de números
de versão de pacotes
no Debian

AVISO

As mini aulas são voltadas para quem já sabe um pouco sobre empacotamento e tratam sobre técnicas específicas.

**Se você deseja aprender a empacotar, comece pelas aulas completas, disponíveis em
<http://debianet.com.br>**

**Aproveite e conheça os meus UDDs em
<https://people.debian.org/~eriberto/udd/>**

Empacotamento de Software no Debian

Mini Aula
Formatos de números
de versão de pacotes
no Debian

Mini Aula

Formatos de números de versão de pacotes no Debian

- **Upstream e revisão Debian**
- **Precedência de símbolos em números de versão**
- **Testes de precedência**
- **Formatos especiais no Debian**
- **Backports**
- **Stable e Old Stable**
- **Exemplo monstro**

Empacotamento de Software no Debian

Mini Aula
Formatos de números
de versão de pacotes
no Debian

Upstream e revisão Debian

- **Os números de versão utilizados pelo Debian para pacotes não-nativos incluem a versão do upstream e a revisão Debian.**
- **A revisão Debian serve para mostrar quantas vezes a mesma versão do código do upstream foi enviado para Debian com mudanças na parte de empacotamento.**
- **A revisão Debian sempre virá depois de um traço (hífen). Exemplo: 2.1-10.**
- **Pacotes nativos não utilizam revisão Debian. Exemplo: 2.10.**

Empacotamento de Software no Debian

Mini Aula
Formatos de números
de versão de pacotes
no Debian

Precedência de símbolos em números de versão

- **Normalmente encontramos dígitos, traços e pontos em números de versão.**
- **Exemplos: 0.1 2.1.11 20200601 1.5-beta**
- **É comum no Debian o uso dos símbolos + e ~ para fazer manobras de precedência em relação ao upstream.**
- **Em números de versão, + é superior e ~ é inferior.**
- **Exemplo: 10.1~qualquercoisa < 10.1 < 10.1+qualquercoisa.**

Empacotamento de Software no Debian

Mini Aula
Formatos de números
de versão de pacotes
no Debian

Testes de precedência

- **Há dois comandos que permitem testar precedência:**
`dpkg --compare-versions` e `sort -V`.
- **O `dpkg --compare-versions` pode ser usado com `lt` (less than), `gt` (greater than) e outros operadores.**
- **O `dpkg --compare-versions` retornará 0 se a condição for verdadeira. Então, necessitaremos do operador Bash `&&` para verificar isso.**
- **Exemplo:** `dpkg --compare-versions 0.1 lt 0.2 && echo ok`

Empacotamento de Software no Debian

Mini Aula
Formatos de números
de versão de pacotes
no Debian

Testes de precedência

- **O comando `sort -v` também pode ser usado. Considere a lista:**

```
10.1  
10.1+qualquercoisa  
10.1~qualquercoisa
```

- **Veja o resultado:**

```
$ sort -V lista2  
  
10.1~qualquercoisa  
10.1  
10.1+qualquercoisa
```

Empacotamento de Software no Debian

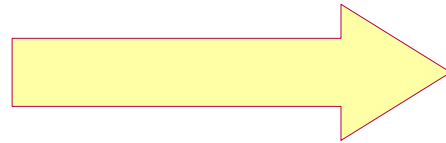
Mini Aula
Formatos de números
de versão de pacotes
no Debian

Testes de precedência

- Mais um exemplo com o antes e o depois:

`$ cat lista`

10.1
10.1+1
10.1~1
10.1-1
10.1-1~1
10.1-1+1
10.1.1
2.1
2.1-1
2.1-1~any



`$ sort -V lista`

2.1
2.1-1~any
2.1-1
10.1~1
10.1
10.1+1
10.1-1~1
10.1-1
10.1-1+1
10.1.1

Empacotamento de Software no Debian

Mini Aula
Formatos de números
de versão de pacotes
no Debian

Testes de precedência

- **Datas podem ser utilizadas em números de versão, desde que iniciem pelo ano, seguido pelo mês, seguido pelo dia, sempre usando dígitos. Assim, será possível haver sequenciamento.**

Exemplo:

20200530

20200531

20200601

20200602

Empacotamento de Software no Debian

Mini Aula
Formatos de números
de versão de pacotes
no Debian

Formatos especiais no Debian

- **O Debian utiliza as regras vistas anteriormente para gerenciar formatos especiais.**
- **Para ver exemplos, execute o comando “dpkg -l” no seu Debian.**
- **A partir de agora veremos vários desses formatos especiais.**
- **Será usado nos exemplos o software fictício conhecido como xyz.**

Empacotamento de Software no Debian

Mini Aula
Formatos de números
de versão de pacotes
no Debian

Formatos especiais no Debian - SVN

- **Commits em repositórios SVN são sequenciais (1, 2, 3 etc).**
- **Assim, é possível inserir no Debian um software que possua uma versão de release pública e commits posteriores.**
- **Considere a versão 2.1 do xyz lançada em certa data e os commits 290, 291 e 292 criados depois do lançamento.**
- **Ao buscarmos todo o software até o commit 292 do upstream, teremos xyz_2.1+svn292-1.**
- **Por outro lado, se tivermos a certeza de que o upstream vai lançar a versão 2.2, poderemos ter xyz_2.2~svn292-1 (arriscado, cuidado).**

Empacotamento de Software no Debian

Mini Aula
Formatos de números
de versão de pacotes
no Debian

Formatos especiais no Debian - SVN

- **Alguns empacotadores usam a data do commit, mas isso é indesejável, pois poderemos ter vários commits no mesmo dia.**
- **Se o upstream nunca lançou uma versão e só trabalha com commits, deveremos usar a versão 0.0. Exemplo:
xyz_0.0+svn292-1 ou xyz_0.0~svn292-1 (a primeira opção é mais utilizada).**

Empacotamento de Software no Debian

Mini Aula
Formatos de números
de versão de pacotes
no Debian

Formatos especiais no Debian - Git

- **O uso de commits Git é similar ao SVN. A diferença é que o número de commit de Git é um hash, ou seja, não é sequencial.**
- **Para Git, deveremos utilizar a data (numérica, iniciada pelo ano) e o número de commit (com 7 caracteres), geralmente separados por um ponto. Exemplos:**

`xyz_0.0+git20200117.cf45a12-1`

`xyz_3.4.14+git20200601.d8a717a-14`

Empacotamento de Software no Debian

Mini Aula
Formatos de números
de versão de pacotes
no Debian

Formatos especiais no Debian - +dfsg

- **DFSG significa Debian Free Software Guidelines.**
- **O sufixo +dfsg é utilizado sempre que um tarball de upstream tem que ser modificado para a retirada de arquivos não livres.**
- **O símbolo + é obrigatório para fazer com que o Debian entenda que o novo tarball é superior a um que já exista no Debian com a mesma versão.**
- **Exemplo: xyz_2.5+dfsg-1**

Empacotamento de Software no Debian

Mini Aula
Formatos de números
de versão de pacotes
no Debian

Formatos especiais no Debian - +dfsg

- **Alguns empacotadores utilizam `dfsg1`, para indicar que é a primeira vez que algo é retirado do tarball. Exemplo:
`xyz_2.5+dfsg1-1`.**
- **De qualquer forma, em uma futura necessidade de remover algo não livre do tarball, `+dfsg2` sempre será superior a `+dfsg` e `+dfsg1`.**

Empacotamento de Software no Debian

Mini Aula
Formatos de números
de versão de pacotes
no Debian

Formatos especiais no Debian - +ds

- **ds** significa **Debian Source**.
- A sua utilização se dá quando alteramos o tarball do upstream para retirarmos bibliotecas de terceiros embutidas (embedded libraries) ou quando removemos um diretório debian/ criado pelo upstream (isso não é mais necessário atualmente).
- **Exemplo:** `xyz_2.5+ds-1`

Empacotamento de Software no Debian

Mini Aula
Formatos de números
de versão de pacotes
no Debian

Formatos especiais no Debian - NMU

- **NMU significa Non-Maintainer Upload.**
- **Um NMU ocorre quando estamos revisando um pacote que possui um mantenedor (não está órfão).**
- **As regras para NMU podem ser encontradas no Debian Developer's Reference.**
- **Para fazer um NMU, use o comando `dch --nmu`.**
- **No NMU em pacotes não-nativos, a revisão Debian anterior, considerando que a mesma não seja NMU, será acrescida de um decimal.**

Empacotamento de Software no Debian

Mini Aula
Formatos de números
de versão de pacotes
no Debian

Formatos especiais no Debian - NMU

- **Exemplo de NMU em pacotes não-nativos: xyz_2.5-5.1**
- **Exemplo de novo NMU no mesmo pacote: xyz_2.5-5.2**
- **Exemplo de NMU provocando “New Upstream Release” (caso raro, evite isso): xyz_2.6-0.1**
- **Para pacotes nativos, será acrescentado o sufixo +nmu1 ou +nmu2 e assim por diante.**
- **Exemplo: xyz_2.5+nmu4**

Empacotamento de Software no Debian

Mini Aula
Formatos de números
de versão de pacotes
no Debian

Formatos especiais no Debian - epoch e really

- **Epoch é um recurso utilizado quando há erros na sequência dos números de versão do upstream ou quando ele muda o esquema de versionamento.**
- **Exemplo: upstream lança as versões 20200101-beta, 20200211-beta, 20200416-beta e, repentinamente, passa a lançar 1.0, 2.0, 2.5. Note que as três primeiras versões são superiores, em número, que as três últimas.**
- **Considerando que todas as versões 2020* já estejam no Debian, deveremos usar o epoch 1: a partir de 1.0. Exemplo: 1:1.0, 1:2.0, 1:3.0.**

Empacotamento de Software no Debian

Mini Aula
Formatos de números
de versão de pacotes
no Debian

Formatos especiais no Debian - epoch e really

- **Caso o upstream altere novamente o esquema de numeração, talvez seja necessário passar a utilizar o epoch 2:.**
- **Epochs não devem ser utilizados para corrigir erros de empacotamento (necessidade de retornar o número de versão, por exemplo). Nesse caso, deveremos usar +really.**
- **Exemplo: xyz_2.6+really2.5-1**

Empacotamento de Software no Debian

Mini Aula
Formatos de números
de versão de pacotes
no Debian

Formatos especiais no Debian - binNMU

- **Eventualmente, o time FTP Master pode comandar um rebuild de um pacote no repositório, a fim de que o mesmo seja reconstruído com uma nova versão de uma dependência crítica.**
- **Outra possibilidade é a falha de construção por erro do ambiente de construção no Debian.**
- **O nome técnico dessa operação é binary NMU (binNMU).**
- **Nesse caso, somente no repositório e nos pacotes oriundos dele, veremos o sufixo +b1, +b2 etc.**
- **Exemplo: xyz_2.6-1+b1**

Empacotamento de Software no Debian

Mini Aula
Formatos de números
de versão de pacotes
no Debian

Formatos especiais no Debian - conflito com upstream

- **Sabemos que o hífen no Debian simboliza o número da revisão. Assim sendo, caso o upstream também use hífen, deveremos renumerar a versão.**
- **Alguns exemplos e possíveis soluções:**
 - **Upstream: 2.1-alpha Debian: 2.1~alpha**
 - **Upstream: pre-2.1 Debian: 2.1~pre**
 - **Upstream: 2-1 Debian: 2.1**
 - **Upstream: 2.1-fixed Debian: 2.1+fixed**

Empacotamento de Software no Debian

Mini Aula
Formatos de números
de versão de pacotes
no Debian

Backports

- **No backports deveremos usar `~bpo<versão Debian>`.**
- **Exemplo de backports para Debian 10: `xyz_2.5-1~bpo10+1`**
- **No caso anterior, `-1` é a revisão original do pacote no testing (origem do backports) e `+1` é a revisão de backports.**
- **Quando `xyz_2.5-1` migrar do testing para o stable, ele substituirá o `xyz_2.5-1~bpo10+1`.**
- **Use `dch --bpo`.**
- **Para saber mais sobre Backports, veja <https://backports.debian.org>.**

Empacotamento de Software no Debian

Mini Aula
Formatos de números
de versão de pacotes
no Debian

Stable e Old Stable

- **Os dois usam o mesmo esquema de numeração.**
- **Considerando o Debian 10 como sendo o atual stable, um upload para stable seria `xyz_2.5-1+deb10u1`.**
- **Cada upload da mesma revisão original terá um incremento após a letra “u”. Exemplo: `xyz_2.5-1+deb10u2`.**
- **Use `dch --stable`.**
- **Antigamente (até o Debian 6), era utilizado o codinome do stable. Exemplo: `xyz_2.5-1+squeeze1`.**

Empacotamento de Software no Debian

Mini Aula
Formatos de números
de versão de pacotes
no Debian

Exemplo monstro

- **Decifre isso, sem a minha ajuda:**

`xyz_2:10.12~beta+dfsg3-12.2~bpo10+2`

Empacotamento de Software no Debian

Mini Aula
Formatos de números
de versão de pacotes
no Debian

FIM!

Eriberto - Jun. 20