

Empacotamento de Software no Debian

Mini Aula Testes de Integração Contínua (CI)

(maio de 2020)

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

Empacotamento de Software 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

Testes de Integração Contínua (CI)

- **Integração Contínua (CI)**
- **Como obter exit 0 quando ele não vem**
- **Pacotes com executáveis em linha de comando**
- **Exemplos em linha de comando**
- **Pacotes com executáveis interativos e GUI**
- **Exemplos interativos e GUI**
- **CI no Salsa**
- **CI no GitHub (curiosidade!)**

Empacotamento de Software no Debian

Integração Contínua (CI)

- **Forma de testar continuamente os programas que estão em um determinado ambiente em constante mudança.**
- **No caso do Debian, testaremos pacotes.**
- **No Debian: `apt install autopkgtest`**
- **A ideia principal é executar os programas, de todas as formas possíveis e imagináveis, continuamente. (é o sonho)**
- **A execução deve ser perfeita (exit 0).**
- **Referência: `/usr/share/doc/autopkgtest/` ---> READMEs**

Empacotamento de Software no Debian

Integração Contínua (CI)

- **Os testes serão realizados no Debian após cada upload e no Salsa após cada push ou commit local.**
- **Eles também serão realizados se as dependências do pacote forem atualizadas (a integração é contínua!).**
- **Desde o lançamento do Debian 10 (Buster), falha de CI gera bug RC[1].**

[1] <https://lists.debian.org/debian-devel-announce/2019/07/msg00002.html>

Empacotamento de Software no Debian

Como obter exit 0 quando ele não vem

- **Veja:**

```
echo ok; echo $?  
ok  
0
```

```
df --lixo; echo $?  
df: unrecognized option '--lixo'  
Try 'df --help' for more information.  
1
```

```
df --lixo | grep Try; echo $?  
df: unrecognized option '--lixo'  
Try 'df --help' for more information.  
1
```

```
df --lixo 2>&1 | grep Try; echo $?  
Try 'df --help' for more information.  
0
```

Empacotamento de Software no Debian

Como obter exit 0 quando ele não vem

- **Em Shell Bash e Dash:**
 - **1 = stdout**
 - **2 = stderr**
- **Então:**
 - **2>1 desvia os erros para o arquivo 1.**
 - **2>&1 desvia os erros para o file descriptor 1 (stdout).**
- **No caso anterior continuaremos a ter erros (saída 1) mas poderemos filtrar uma expressão qualquer com grep.**

Empacotamento de Software no Debian

Como obter exit 0 quando ele não vem

- `2>&1 | grep` irá capturar o resultado final.
- Se não chegarmos ao final, o resultado não será 0! Exemplos: buffer overflow e segmentation fault.
- From https://en.wikipedia.org/wiki/Segmentation_fault:

```
int main(void)
{
    char *s = "hello world";
    *s = 'H';
}

$ gcc segfault.c -g -o segfault
$ ./segfault
```


Empacotamento de Software no Debian

Como obter exit 0 quando ele não vem

- **É comum as páginas de help (-h, --help) e as indicações de versão (-v, -V, --version) produzirem saída 1.**

Empacotamento de Software no Debian

Pacotes com executáveis em linha de comando

- **Crie o arquivo** `debian/tests/control`.
- **Coloque uma linha** `Test-Command:` **para testes diretos. Para mais de um teste, deverá haver uma linha em branco separando. Exemplo:**

```
Test-Command: dd --version
```



```
Test-Command: dd if=debian/tests/file.txt of=$AUTOPKGTEST_TMP/ok.txt
```
- **Para executar um script (Bash, Perl, Python) como teste, use** `Tests:.` **O script deverá estar em** `debian/tests/.` **Exemplo:**

```
Tests: testando.sh
```

Empacotamento de Software no Debian

Pacotes com executáveis em linha de comando

- **Podrá haver dependências e restrições.**
- **Dependências são pacotes necessários para realizar o teste.**
- **São sinalizadas pelo campo Depends.**
- **O nome do pacote pode ser substituído por @. Exemplo:**
Depends: @, apache2
- **O pacote sempre será instalado no momento do teste de CI.
Se ele for a única dependência, o campo Depends não
precisará existir.**

Empacotamento de Software no Debian

Pacotes com executáveis em linha de comando

- **As restrições são condições especiais exigidas por algum teste.**
- **Exemplo: os testes sempre rodam sobre usuário comum.**
Caso haja necessidade, deverá ser acrescentada uma restrição para que ele rode como root.
- **As restrições são viabilizadas pelo campo Restrictions.**
Exemplo:
`Restrictions: allow-stderr, needs-root`
- **Veja as restrições em `/usr/share/doc/autopkgtest/README.package-tests.html`**

Empacotamento de Software no Debian

Pacotes com executáveis em linha de comando

- **No Debian e no Salsa, os testes são executados em VMs ou Docker.**
- **Para executar localmente:**
 - **Construa o pacote (debuild).**
 - **Instale o pacote (debi).**
 - **Alterne para um usuário comum.**
 - **No diretório do upstream, use: `$ autopkgtest . -- null`**

Empacotamento de Software no Debian

Exemplos em linha de comando

- **detox**
- **jdupes**
- **magicrescue**
- **axel**

Empacotamento de Software no Debian

Pacotes com executáveis interativos e GUI

- **Não existe um método específico para fazer testes dentro de um ambiente gráfico ou em ambientes interativos em shell.**
- **Solução: xvfb (X Virtual Frame Buffer).**
- **O xvfb funciona como se fosse um ambiente gráfico, rodando em memória RAM, sem stdout/stderr.**
- **Com ele é possível executar programas e verificar se há falhas de segmentação e outras ocorrências indesejáveis.**

Empacotamento de Software no Debian

Mini Aula
CI Tests

Exemplos interativos e GUI

- **qabcs**
- **hexedit**
- **netdiscover**

Empacotamento de Software no Debian

CI no Salsa

- **O servidor Salsa é o repositório Git do Debian. Nele é possível fazer o controle de versão do empacotamento.**
- **O Salsa possui um sistema de CI que faz vários testes sobre o pacote automaticamente.**
- **Também é possível fazer testes personalizados.**

Empacotamento de Software no Debian

CI no Salsa

- **Para que isso aconteça é necessário criar um arquivo `debian/salsa-ci.yml`, com o seguinte conteúdo:**

```
---
```

```
include:
```

- `https://salsa.debian.org/salsa-ci-team/pipeline/raw/master/salsa-ci.yml`
- `https://salsa.debian.org/salsa-ci-team/pipeline/raw/master/pipeline-jobs.yml`

- **Você poderá copiar isso de `/usr/share/debhelper/dh_make/debian/salsa-ci.yml.ex`, caso você tenha o pacote `dh-make` instalado.**

Empacotamento de Software no Debian

CI no Salsa

- **A seguir, é preciso entrar no Salsa e configurar em cada repositório desejado:**
 - **Settings > CI/CD > Expand (em General pipelines).**
 - **Em "Custom CI config path" escreva `debian/salsa-ci.yml`**
 - **Clique em "Save Changes".**
 - **Os testes vão rodar depois do próximo push para o Salsa ou commit diretamente nele.**
- **O CI do Salsa elimina vários passos “manuais” do checklist normal de um empacotador.**

Empacotamento de Software no Debian

CI no Salsa

- **É possível permitir a falha de algum teste e desabilitar outros. Exemplo:**

```
---
```

```
include:
```

- <https://salsa.debian.org/salsa-ci-team/pipeline/raw/master/salsa-ci.yml>
- <https://salsa.debian.org/salsa-ci-team/pipeline/raw/master/pipeline-jobs.yml>

```
reprotest:
```

```
  allow_failure: true
```

```
variables:
```

```
  SALSA_CI_DISABLE_BLHC: 1
```

Empacotamento de Software no Debian

CI no GitHub (curiosidade!)

- **O GitHub também permite rotinas de CI para cada push (incluindo os Push Requests).**
- **O GitHub chama isso de Actions.**
- **As Actions podem ser executadas em Linux (Ubuntu), Windows e macOS.**
- **Exemplo em [/resurrecting-open-source-projects/dcfldd](#) (GitHub).**