**Tutorial para criação de pacotes debian para o Donnie**

Primeiramente, é preciso criar um diretório base para montar a estrutura do pacote. Por exemplo:

$> mkdir tmp/pacote-donnie-player-stage-0.1

Dentro do diretório base crie um diretório DEBIAN (**deve ser em maiúsculo**):

$> cd tmp/pacote-donnie-player-stage-0.1

$> mkdir DEBIAN

Neste diretório deve ser criado um arquivo chamado “control” (sem nenhuma extensão). Use o editor de texto de sua preferência, neste caso foi usado o Geany:

$> cd DEBIAN

$> geany control &

O arquivo control contém informações importantes sobre o pacote. Segue abaixo os principais campos:

* Package: nome do pacote
* Version: versão do pacote que será criado
* Architecture: lista de arquitetura(s) para a(s) qual(is) o pacote é destinado
* Maintainer: nome do mantenedor do pacote <mantenedor@email.com> (email é opcional)
* Depends: lista de dependências para instalação do pacote (não deve haver quebra de linha)
* Description: descrição sobre o pacote

Mais informações sobre os campos do arquivo control no link: <https://www.debian.org/doc/debian-policy/ch-controlfields.html#s-controlsyntax>

Exemplo de arquivo control:

|  |
| --- |
| Package: donnie-player-stage  Priority: optional  Version: 0.1  Architecture: amd64  Maintainer: Joice Marek  Depends: libsox2, libsox-fmt-all, libreadline5, libantlr3c-3.2-0, libfltk1.1, freeglut3, libpng12-0, libltdl7, libdb5.1-stl, libgnomecanvasmm-2.6-1c2a, python-gnome2, libboost-thread1.54.0, libboost-signals1.54.0, libboost-system1.54.0, libopencv-core2.4, libopencv-highgui2.4, libcv2.4, libcvaux2.4, libasound2, libgsl0ldbl, libxmu6, libpq5, libpqxx-3.1  Description: Donnie is an inclusive robotic programming environment which can be used by all students, with or without visual disabilities.  Homepage: https://github.com/lsa-pucrs/donnie-assistive-robot-sw/ |

**Arquivo control do pacote Donnie versão 0.1**

Outros arquivos que podem ser criados no diretório DEBIAN são os scripts postinst, preinst, postrm e prerm (sem extensão). Estes scripts são executados antes/depois da instalação/remoção do pacote. Para o pacote do Donnie 0.1 foram utilizados apenas o postinst e o postrm.

|  |
| --- |
| #!/bin/bash  export DONNIE\_PATH="/opt/donnie"  echo "source $DONNIE\_PATH/setup.sh" >> ~/.bashrc |

**postinst**

|  |
| --- |
| #!/bin/bash  #remove do ~/.bashrc a linha 'source /opt/donnie/setup.sh'  sed -i.bak '\~source /opt/donnie/setup.sh~d' ~/.bashrc |

**postrm**

Voltando ao diretório base, devemos criar a estrutura de diretórios do pacote que desejamos após a instalação. O diretório base representa o diretório root, por isso a estrutura deve ser criada a partir dele.

Sendo assim, com o Donnie instalado no seu computador, copie a estrutura de diretórios onde ele foi instalado para o diretório base. O pacote do Donnie 0.1 também instala o Player e o Stage, então faça o mesmo para eles.

$> cp -r /opt/donnie/ tmp/pacote-donnie-player-stage-0.1/opt/donnie/

$> cp -r /opt/player/ tmp/pacote-donnie-player-stage-0.1/opt/player/

$> cp -r /opt/stage/ tmp/pacote-donnie-player-stage-0.1/opt/stage/

A estrutura do pacote deve ser semelhante à esta:

|  |
| --- |
| pacote-donnie-player-stage-0.1/  ├── DEBIAN/  │ ├── control  │ ├── postinst  │ └── postrm  └── opt/  ├── donnie/  │ ├── bin/  │ ├── docs/  │ ├── lib/  │ ├── resources/  │ ├── setup.sh  │ └── test/  ├── player/  │ ├── bin/  │ ├── include/  │ ├── lib64/  │ └── share/  └── stage/  ├── bin/  ├── include/  ├── lib64/  └── share/ |

**Estrutura do pacote Donnie 0.1**

Com os arquivos do diretório DEBIAN e a estrutura prontos, podemos criar o pacote debian com o comando:

$> dpkg-deb -b tmp/pacote-donnie-player-stage-0.1 tmp

Obs.: dpkg-deb -b <caminho para o diretório base> <diretório onde será criado o pacote>

Para instalar o pacote use os seguintes comandos:

$> sudo apt-get update

$> sudo dpkg -i donnie-player-stage\_0.1\_amd64.deb

Caso a instalação apresente uma mensagem de erro de dependências, use o comando abaixo:

$> sudo apt-get -f install