Crash Course ROS2 e Máquina virtual

Tópicos Especiais em Robótica

Prof. Roberto de Souza Baptista

Alunos: Rodrigo Venturini de Lima 19/0095172

Rafael Brasileiro Vanderlei Rodrigues da Costa 19/0044110

Sumário

1. Baixar Virtualbox e Iso do Ubuntu 22.04 (Jammy Jellyfish)	
1.1. Instalar Virtualbox	4
2. Como fazer uma máquina virtual a partir da iso	7
3. Como instalar o ROS 2 na máquina	8
3.1. Setup sources	8
3.2. Instalar os pacotes do ROS 2	8
3.3. Configuração de ambiente	9
3.4. Configurar o ROS 2	9
4. Aprendendo o ROS 2	11
4.1. Utilizando múltiplos nodes	11
4.2. Criando um ambiente de trabalho do ROS 2	12
4.3. Fazendo o primeiro Node	15

1. Baixar Virtualbox e Iso do Ubuntu 22.04 (Jammy Jellyfish)

Para baixar o virtual box acessar: https://www.virtualbox.org/wiki/Downloads (fig.1) e https://releases.ubuntu.com/jammy/ (fig.2),com isso é só clicar aonde a seta aponta fazer o download: do instalador da Virtual Box(se for Windows, se for outro sistema operacional clicar o específico) e da ISO do Ubuntu 22.04.

Obs.: a Iso do Ubuntu só será utilizado como uma imagem, então não inicializar ela

WirtualBox Download VirtualBox Here you will find links to VirtualBox binaries and its source code. VirtualBox binaries By downloading, you agree to the terms and conditions of the respective license. VirtualBox 7.1.0 platform packages Windows hosts Windows hosts

Fig.1

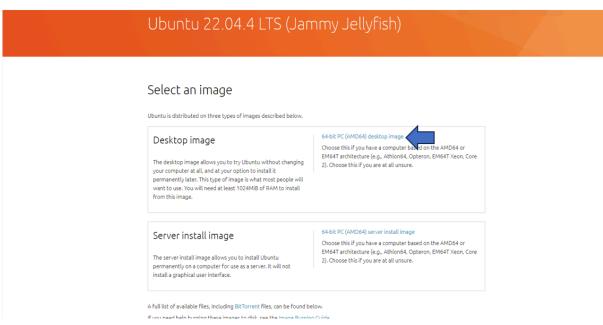


Fig.2

1.1. Instalar Virtualbox

Quando terminar o download do instalador do Virtualbox no instalador que estará em download e com o nome de "VirtualBox-7.1.0-164728-Win" (claro se não alterar o nome do instalador) (fig.3) e em seguida aparecerá (fig.4), para prosseguir aperte em "Next>"ç



(fig.4)

Em Custom setup (fig.5) pode deixar o padrão e clicar e "Next>" e fazer o mesmo com a (fig.6) mas em vez de "Next>", clicar "yes";

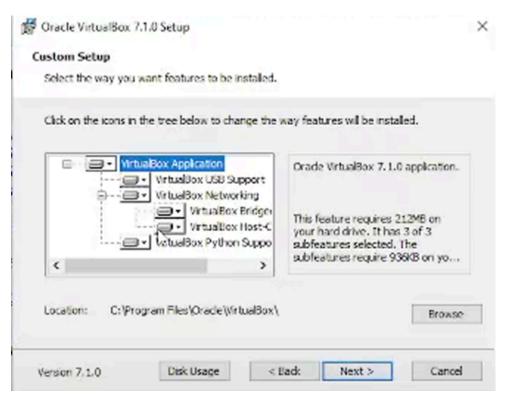


fig.5



Fig. 6

Na (fig.7) clicar em "Next>" e deixar tudo marcado na (fig.8) e clicar "Next>";

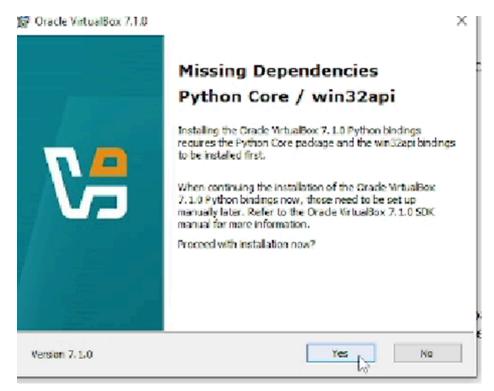


fig.7

🙀 Oracle VirtualBox 7.1.0 Setup		>
Custom Setup		
Select the way you want feature	s to be installed.	
Please choose from the options b	oclose:	
Create start menu entries		
Create a shortcut on the des	ktop	
Create a shortcut in the Quid	kLaunch Bar	
☑ Register file associations		
	₽-	
Version 7.1.0	< Back Next >	Cancel

fig.8

Na (fig.9) tem que clicar em "install" para finalizar a instalação e quando aparecer a (fig.10) apertar em "finish" para finalizar a instalação.

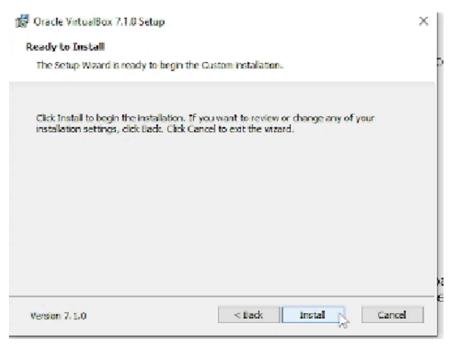


Fig. 9



Fig. 10

2. Como fazer uma máquina virtual a partir da iso

Acessar o link: https://youtu.be/DlpZqza-dRw

3. Como instalar o ROS 2 na máquina

Para este Guia foi usada a versão Humble do ROS 2, caso deseje usar outra versão, no site https://docs.ros.org/ pode se encontrar o guia de instalação de todas as versões.

3.1. Setup sources

Primeiramente é necessário adicionar o repositório do ROS 2 no sistema

3.1.1. Primeiro certifique-se que o Repositório Universal do Ubuntu está habilitado colocando os comandos no terminal de comando

sudo apt install software-properties-common

sudo add-apt-repository universe

3.1.2. Agora adicione a chave GPC do ROS 2 com o apt

sudo apt update && sudo apt install curl -y

sudo curl -sSL https://raw.githubusercontent.com/ros/rosdistro/master/ros.key -o /usr/share/keyrings/ros-archive-keyring.gpg

3.1.3. Agora adicione o repositório na sua source list

echo "deb [arch=\$(dpkg --print-architecture) signed-by=/usr/share/keyrings/ros-archive-keyring.gpg] http://packages.ros.org/ros2/ubuntu \$(. /etc/os-release && echo \$UBUNTU_CODENAME) main" | sudo tee /etc/apt/sources.list.d/ros2.list > /dev/null

3.2. Instalar os pacotes do ROS 2

3.2.1. Atualize seu sistema usando os comandos

sudo apt update

sudo apt upgrade

3.2.2. Agora faça as instalações necessárias do ROS 2 com os comandos

sudo apt install ros-humble-desktop

sudo apt install ros-humble-ros-base

sudo apt install ros-dev-tools

3.3. Configuração de ambiente

Por fim, só usar o comando do source no seguinte arquivo para habilitar o ROS 2 no terminal (será necessário fazer isso em todo terminal que for aberto, mas vamos resolver isso mais para frente).

source /opt/ros/humble/setup.bash

3.3.1. Teste um exemplo para ver o ROS 2 funcionando

source /opt/ros/humble/setup.bash

ros2 run demo_nodes_cpp talker

Este exemplo fará que uma mensagem comece a aparecer no terminal e um número ao final dela que aumentará em 1 a cada mensagem:

```
[INFO] [1726490232.339708617] [talker]: Publishing: 'Hello World: 1'
[INFO] [1726490233.339999621] [talker]: Publishing: 'Hello World: 2'
[INFO] [1726490234.340006398] [talker]: Publishing: 'Hello World: 3'
[INFO] [1726490235.339986920] [talker]: Publishing: 'Hello World: 4'
[INFO] [1726490236.339590004] [talker]: Publishing: 'Hello World: 5'
```

Após isso o ROS 2 estará pronto para a utilização. Para parar o programa só apertar Control + C no terminal.

3.4. Configurar o ROS 2

Para não precisar ter que digitar o comando source citado no item 3.3. toda vez que abrir um terminal para habilitar o ROS 2 é necessário fazer o seguinte:

3.4.1. No terminal digite o comando

gedit ~/.bashrc

Abrirá um arquivo de texto no computador como mostrado na figura 11.

```
.bashrc
  Open Y
                                                                                                                                                                                             1# ~/.bashrc: executed by bash(1) for non-login shells.
2 # see /usr/share/doc/bash/examples/startup-files (in the package bash-doc)
3 # for examples
  5 # If not running interactively, don't do anything
 6 case $- in
7 *i*);;
8 *) return;;
11 # don't put duplicate lines or lines starting with space in the history.
12 # See bash(1) for more options
13 HISTCONTROL=ignoreboth
15 # append to the history file, don't overwrite it
16 shopt -s histappend
18 # for setting history length see HISTSIZE and HISTFILESIZE in bash(1)
19 HISTSIZE=1000
20 HISTFILESIZE=2000
21 # check the window size after each command and, if necessary, 23 # update the values of LINES and COLUMNS.
24 shopt -s checkwinsize
26 # If set, the pattern "**" used in a pathname expansion context will 
27 # match all files and zero or more directories and subdirectories.
28 #shopt -s globstar
30 # make less more friendly for non-text input files, see lesspipe(1)
31 [ -x /usr/bin/lesspipe ] && eval "$(SHELL=/bin/sh lesspipe)"
32 # set variable identifying the chroot you work in (used in the prompt below)
34 if [ -z "${debian_chroot:-}" ] && [ -r /etc/debian_chroot ]; then
35 debian_chroot=$(cat /etc/debian_chroot)
36 fi
```

Fig. 11

3.4.2. Vá até a última linha do arquivo e coloque o comando a seguir como mostra a figura 12.

source /opt/ros/humble/setup.bash

Fig 12

- 3.4.3. Salve e feche o arquivo
- 3.4.4. Abra um novo terminal de comando e tente rodar o programa exemplo do item 3.3.1 sem o comando source e verá que ele funcionará normalmente.

Obs: O terminal utilizado ao longo do tutorial será o "Tabby", mas tudo que será feito pode ser utilizado no terminal base do Ubuntu.

4. Aprendendo o ROS 2

4.1. Utilizando múltiplos nodes

Nodes são programas do ROS 2. Como visto no item 3.1.1, ao instalar o ROS 2, também são instalados alguns nodes exemplo. Além do node que já foi testado também existe um node que comunica com ele.

Abra um terminal e ative o node:

```
ros2 run demo nodes cpp talker
```

E em outro terminal ative o node:

ros2 run demo_nodes_cpp listener

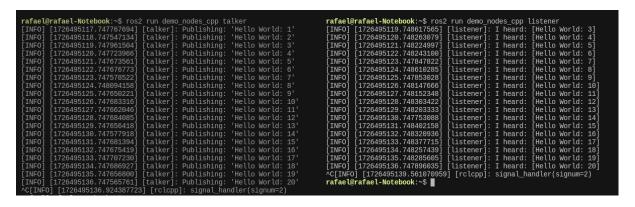


Fig.13

Abrindo um novo terminal e digitando o comando

rqt_graph

É possível ver a comunicação entre os dois nodes como é visto na figura 14:

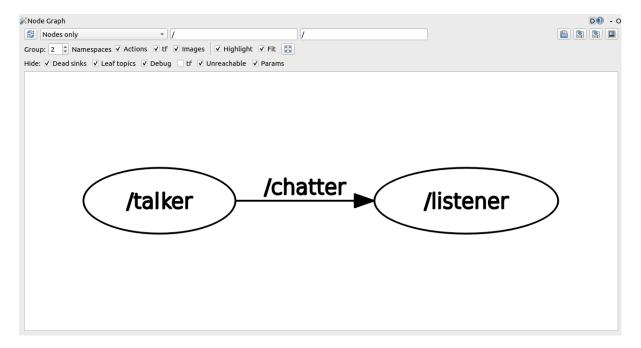


Fig. 14

/talker é o node que foi iniciado primeiro que publica a mensagem "Hello World", enquanto o /listener vai receber a mensagem que foi publicada pelo outro node. Por fim existe o /chatter que não é um node mas sim um topic. Topic é uma forma de comunicação entre nodes, portanto /talker não está comunicando diretamente com /listener, mas com /chatter.

4.2. Criando um ambiente de trabalho do ROS 2

Um ambiente de trabalho é uma área para organizar os códigos e poder compartilhar com outras pessoas o trabalho.

4.2.1. Instalar Colcon

Primeiro Instale o colcon para poder criar nodes para o ROS 2 com o comando:

sudo apt install python3-colcon-common-extensions

E complete a instalação do colcon. Agora localize a pasta usando cd no terminal:

cd /usr/share/colcon_argcomplete/hook

E use o comando para editar o arquivo bashrc da mesma forma que no item 4.3.1 e coloque na linha de comando abaixo da última linha o comando:

source /usr/share/colcon_argcomplete/hook/colcon-argcomplete.bash

Salve e feche o arquivo. Agora de o comando "cd" para sair da pasta no terminal e usando o comando

source ~/.bashrc

E o colcon já estará disponível para utilização.

4.2.2. Criar um diretório

Para criar um diretório basta usar o comando:

mkdir ros2_ws

O nome para o diretório usado nesse tutorial é ros2_ws, mas pode ser colocado qualquer nome para o diretório que desejar criar. Agora entre na pasta do diretório com o comando:

cd ros2 ws/

Dentro da pasta crie uma pasta chamada src para funcionar como source do diretório.

mkdir src

Agora com o comando:

colcon build

Assim foi montado o ambiente de trabalho para realizar as atividades com ROS 2 de maneira mais organizada. Usando o comando "Is" é possível ver que no diretório agora existem 4 pastas, assim que temos a certeza que o ambiente de trabalho foi montado, como é visto na figura 15.

```
rafael@rafael-Notebook:~$ cd ros2_ws/
rafael@rafael-Notebook:~/ros2_ws$ ls
build install log src
rafael@rafael-Notebook:~/ros2_ws$
```

Figura 15

Agora para tornar funcional o ambiente de trabalho e os nodes que forem criados poderem ser instalados e usados, use o comando "cd" para sair da pasta e o comando

source ~/ros2_ws/install/setup.bash

E após isto edite o arquivo ./bashrc como das outras vezes e coloque após a última linha este último comando source desta forma:

```
119 <mark>source</mark> /opt/ros/humble/setup.bash
120 <mark>source</mark> /usr/share/colcon_argcomplete/hook/colcon-argcomplete.bash
121 source ~/ros2_ws/install/setup.bash
```

Agora o ambiente de trabalho está criado e configurado

4.2.3. Criando um python package

Entre na pasta src que foi criada no último guia usando o comando:

cd ros2 ws/src

E crie o pacote dentro da pasta com o comando:

ros2 pkg create my_robot_controller --build-type ament-python --dependencies rclpy

A partir de agora será preciso um software para programar, a recomendação é o VS Code por se tratar de um software mais universal e que possui compatibilidade como ROS2.

Volte para a pasta do ros2_ws e use o comando "colcon build", caso de algum erro durante a operação do comando siga os comandos a seguir:

sudo apt install python3-pip

pip3 install setuptools==58.2.0

Agora só usar o comando "colcon build" mais uma vez e vai funcionar

4.3. Fazendo o primeiro Node

Primeiro entre na pasta do package que foi criado no último passo:

cd ros2_ws/src/my_robot_controller/my_robot_controller/

E agora crie um arquivo python para ser o código do seu primeiro node:

touch my_first_node.py

chmod +x my_first_node.py

E agora retorne para a pasta src e abra o VS Code usando o comando "Code .", assim o sistema vai abrir o VS Code diretamente na pasta.

Agora vamos criar um código Hello World para ROS 2. Entre no github situado no link https://github.com/RafaelBVRC/RoboticaROS2 e veja o código "My first node.py" lá

Após a criação do node, ele precisa ser instalado dentro do vscode entre no setup.py encontre onde está escrito 'console_scripts' e digite entre os colchetes

"test_node = my_robot_controler.my_first_node:main"

onde test_node será o comando para rodar o node my_first_node

Agora dentro da pasta ros2 ws rode os comandos no terminal

colcon build

source ~/.bashrc

ros2 run my_robot_controller test_node

E o node vai rodar normalmente. Lembrando mais uma vez que para interromper o node deve-se clicar control+c no terminal onde ele está

rondando. Para atualizar pela primeira vez o node depois de alguma alteração no código é necessário atualizar da seguinte forma:

colcon build --symlink-install

source ~/.bashrc

ros2 run my_robot_controller test_node

A partir desse código, toda atualização será automática e precisará apenas dar o source e rodar o node com o ROS 2.

No github https://github.com/RafaelBVRC/RoboticaROS2/ existem outros nodes prontos mais avançados que podem ser estudados para aprimorar o conhecimento no ROS2