Publicação de Mensagens no ROS 2

Walter Fetter Lages

fetter@ece.ufrgs.br

Universidade Federal do Rio Grande do Sul

Escola de Engenharia

Departamento de Sistemas Elétricos de Automação e Energia

ENG10052 Laboratório de Robótica

UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

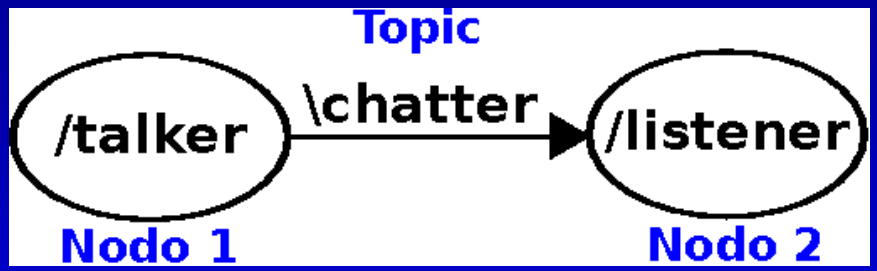
Introdução

Nodo: processo do S.O. hospedeiro

Tópico: mecanismo de comunicação entre nodos do tipo *publisher/subscriber*

Mensagem: dados publicados nos tópicos

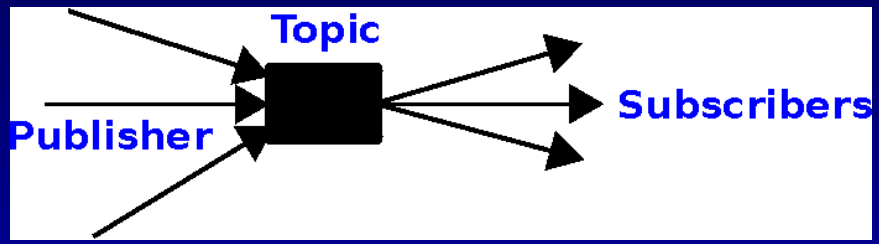
Gráfico de computação: Representa a comunicação entre os nodos através de tópicos



UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Tópicos

- Nodos podem publicar mensagens em tópicos
- Cada tópico pode ter vários publicadores e assinantes



- Cada nodo pode publicar ou assinar vários tópicos
- Publicadores e assinantes não sabem da existência um dos outros
- A ordem de execução não é garantida
- Comunicação assíncrona

UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Turtlesim

- Tartaruga simulando o modelo cinemático de um robô móvel

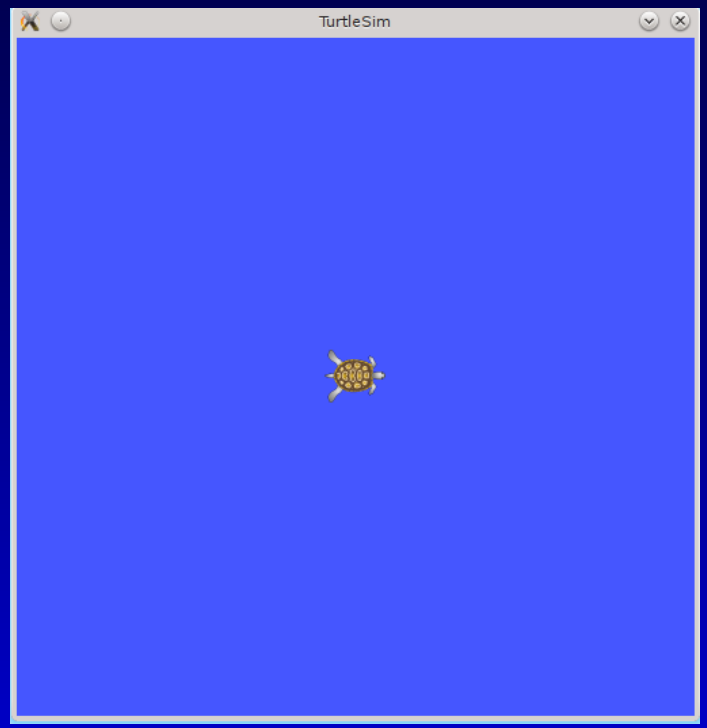
$$\begin{aligned} \dot{x} &= \cos(\theta)v \\ \dot{y} &= \text{sen}(\theta)v \\ \dot{\theta} &= \omega \end{aligned}$$

- Comandada por tópicos e serviços
- Aqui será feito um programa para comandar a tartaruga publicando mensagens
  - Publicar v e ω no tópico assinado pelo `turtlesim`

UFRGS
UNIVERSIDADE FEDERAL
DO RIO GRANDE DO SUL

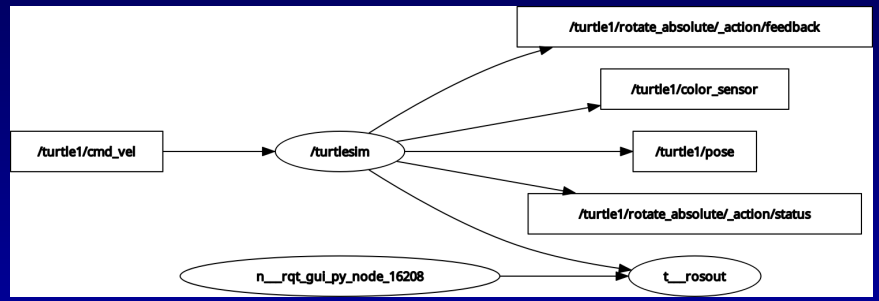
Turtlesim

```
ros2 run turtlesim turtlesim_node &
```



Comando da Tartaruga

- Utilizando-se o `rqt_graph` ou `ros2 topic list` pode-se verificar que o movimento da tartaruga é comandado através do tópico `/turtle1/cmd_vel`



- Movimenta-se a tartaruga publicando neste tópico
- O tipo da mensagem pode ser verificado com o comando:

---

ros2 `topic info /turtle1/cmd_vel`

  - `geometry_msg/msg/Twist`

Mensagem geometry_msg/msg/Twist

- O formato da mensagem pode ser verificada com:

```
ros2 interface show geometry_msgs/msg/Twist
```

```
Vector3  linear
    float64 x
    float64 y
    float64 z
Vector3 angular
    float64 x
    float64 y
    float64 z
```

- Neste caso, apenas a velocidade linear em `x` e a velocidade angular em `z` são efetivamente usados

UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Programa para Comandar a Tartaruga

- O ROS já vem com o programa `turtle_teleop_key`
  - Comanda a tartaruga através do teclado

---

ros2 `run turtlesim turtle_teleop_key`

- Aqui será feito um programa para comandar a tartaruga pelo próprio *software*
  - O objetivo não é o programa em sí, mas o processo de desenvolvimento e os conceitos usados
- Programas do ROS devem ser colocados dentro de pacotes

Estrutura do Pacote no ROS 2

```
colcon_ws/
├── build/
├── install/
├── log/
└── src/
    └── turtle_command/
        ├── CMakeLists.txt
        ├── launch/
        │   ├── turtle_command.launch.py
        │   └── turtle_command.launch.xml
        ├── package.xml
        └── src/
            └── turtle_command.cpp
```

Criação do Pacote

```
cd ~/colcon_ws/src
ros2 pkg create --build-type ament_cmake --
    dependencies rclcpp geometry_msgs
--node-name turtle_command turtle_command
```

```
turtle_command/
├─CMakeLists.txt
├─include/  ......Não será usado, pode ser removido
├─package.xml
└─src/
  └─turtle_command.cpp
```

- Editar o arquivo `package.xml` para configurar os detalhes de documentação e dependências
- Editar o arquivo `CMakeLists.txt` para ajustar os detalhes de compilação do pacote

UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

`package.xml`

- Editar o arquivo `package.xml` para preenchimento dos meta-dados do pacote
  - Descrição
  - Mantenedor
  - Licença
  - Dependências
  - Exportações

UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

CMakeLists.txt

- Editar `CMakeLists.txt` para descomentar e ajustar as *tags*:

```
add_executable(turtle_command
   src/turtle_command.cpp)

ament_target_dependencies(turtle_command
   rclcpp geometry_msgs)

install(TARGETS turtle_command
  DESTINATION lib/${PROJECT_NAME})

install(DIRECTORY launch
  DESTINATION share/${PROJECT_NAME})
```

UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Código Fonte do Nodo

- Editar o arquivo com o código fonte no diretório `src`
- arquivo `turtle_command.cpp`

turtle_command.cpp

```
#include <rclcpp/rclcpp.hpp>
#include <geometry_msgs/msg/twist.hpp>

class TurtleCmd: public rclcpp::Node
{
    public:
    TurtleCmd(void);
    void setCommandCB(void) const;

    private:
    rclcpp::Publisher<geometry_msgs::msg::Twist
    >::SharedPtr cmdPublisher_;
    rclcpp::TimerBase::SharedPtr timer_;
};
```

turtle_command.cpp

```
TurtleCmd::TurtleCmd(void): Node("turtle_command
    ")
{
    cmdPublisher_=create_publisher<geometry_msgs
    ::msg::Twist>("/turtle/cmd",10);

    using namespace std::chrono_literals;
    timer_=rclcpp::create_timer(this,this->
    get_clock(),100ms,std::bind(&TurtleCmd::
    setCommandCB,this));
}
```

turtle_command.cpp

```
void TurtleCmd::setCommandCB(void) const
{
    geometry_msgs::msg::Twist cmd;

    cmd.linear.x=1.0;
    cmd.linear.y=1.0;
    cmd.linear.z=0.0;

    cmd.angular.x=0.0;
    cmd.angular.y=0.0;
    cmd.angular.z=1.0;

    cmdPublisher_->publish(cmd);
}
```

turtle_command.cpp

```
int main(int argc,char* argv[])
{
    rclcpp::init(argc,argv);

    rclcpp::spin(std::make_shared<TurtleCmd>());

    rclcpp::shutdown();
    return 0;
}
```

UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Compilar

- Compilar com os comandos:

```
cd ~/colcon_ws
colcon build --symlink-install
```

- Ou para compilar só o pacote:

```
cd ~/colcon_ws
colcon build --symlink-install  --packages-
    select turtle_command
```

UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Baixar o pacote

- Baixar, descompactar e compilar o pacote com os comandos:

```
cd ~/colcon_ws/src
git clone -b $ROS_DISTRO http://git.ece.
    ufrgs.br/eng10026/turtle_command
cd ~/colcon_ws
colcon build --symlink-install
```

Execução do Nodo

- Chamar diretamente o executável:

```
~/colcon_ws/build/turtle_command/
    turtle_command --ros-args --remap /
    turtle/cmd:=/turtle1/cmd_vel
```

- Usando o comando `ros2 run`:

```
ros2 run turtle_command turtle_command --ros
    -args --remap /turtle/cmd:=/turtle1/
    cmd_vel
```

- Notar o remapeamento no nome dos tópicos
- Usando um arquivo de *launch*:

```
ros2 launch turtle_command turtle_command.
    launch.py
```

UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Arquivos de *Launch*

- Arquivos de *launch* devem estar dentro de pacotes
- No ROS 1 são arquivos em XML
- No ROS 2 usalmente são programas em Python
  - Podem ser em XML ou YAML, mas ainda não tem suporte às novas funcionalidades do ROS 2
- São executados com o comando `ros2 launch`

```
ros2 launch turtle_command turtle_command.
    launch.py
```

```
ros2 launch turtle_command turtle_command.
    launch.xml
```

turtle_command.launch.py

```
from launch import LaunchDescription
from launch_ros.actions import Node

def generate_launch_description():
    return LaunchDescription([
        Node(
            name='turtle',
            package='turtlesim',
            executable='turtlesim_node',
            output='screen'
        ),
```

turtle_command.launch.py

```
        Node(
            name='turtle_cmd',
            package='turtle_command',
            executable='turtle_command',
            remappings=[('/turtle/cmd','/
  turtle1/cmd_vel')],
            output='screen'
        )
    ])
```

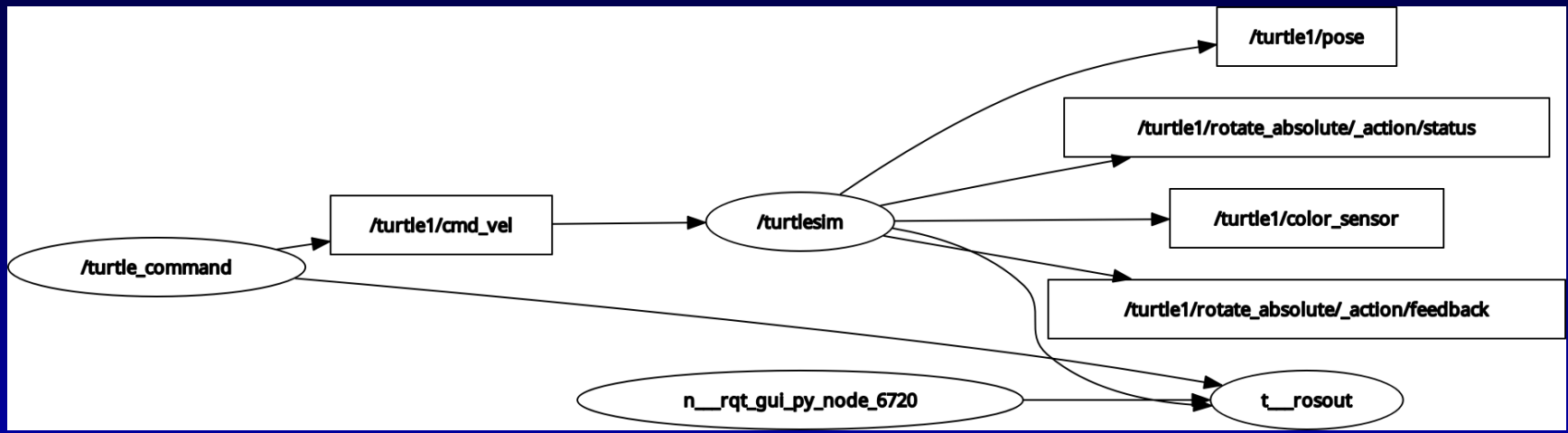
turtle_command.launch.xml

```
<launch>
  <node name="turtle" pkg="turtlesim" exec="
   turtlesim_node"/>

  <node name="turtle_cmd" pkg="turtle_command"
   exec="turtle_command">
     <remap from="/turtle/cmd" to="/turtle1/
   cmd_vel"/>
  </node>
</launch>
```

Gráfico de Computação

rqt_graph &



Pacote turtle_command

```
colcon_ws/
└─src/
  └─turtle_command/
    ├─CMakeLists.txt
    ├─package.xml
    ├─launch/
    │ ├─turtle_command.launch.py
    │ └─turtle_command.launch.xml
    └─src/
      └─turtle_command.cpp
```

UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Outros Comandos para Tópicos

- Vizualizar o que é publicado no tópico

```
ros2 topic echo /turtle1/cmd_vel
```

- Obter a taxa com que mensagens são publicadas

```
ros2 topic hz /turtle1/cmd_vel
```

- Publicar mensagem

```
ros2 topic pub /turtle1/cmd_vel
    geometry_msgs/msg/Twist '{linear: {x:
    1.0}, angular: {z: 1.0} }'
```

UFRGS
UNIVERSIDADE FEDERAL
DO RIO GRANDE DO SUL

Exercícios

- Faça um programa para a tartaruga fazer um 8.
- Use o `rqt_plot` para plotar gráficos da posição e orientação da tartaruga em função do tempo