

Mãos à obra

- **Instalando o ROS e definindo variáveis de ambiente.**
- **Trabalhando com Tópicos, Serviços e Parâmetros no TurtleSim.**
- **Criando um espaço de trabalho (catkin_workspace)**
- **Criando um pacote**
- **Trabalhando com launch**
- Escrevendo um nodo em roscpp que publica e um que se inscreve em um tópico
- Escrevendo um nodo em rospy para o TurtleSim
- Trabalhando com o simulado Stage
- Desenhando no RViz
- Utilizando o ROS em múltiplas máquinas.

Mãos à obra

- Instalando o ROS e definindo variáveis de ambiente.
- Trabalhando com Tópicos, Serviços e Parâmetros no TurtleSim.
- Criando um espaço de trabalho (catkin_workspace)
- Criando um pacote
- Trabalhando com launch
- Escrevendo um nodo em roscpp que publica e um que se inscreve em um tópico
- Escrevendo um nodo em rospy para o TurtleSim
- Desenhando no RViz
- Utilizando o ROS em múltiplas máquinas.

Escrevendo um nodo que publica e um que se inscreve em um tópico

- **Criando o nodo talker (pública)**
 - Anuncia um tópico chamado “chatter”
 - Pública msgs
 - Mostra msgs publicadas

Escrevendo um nodo que publica e um que se inscreve em um tópico

- Criando o nodo talker (pública):

Arquivo: **talker.cpp**

```
#include "ros/ros.h"
#include "std_msgs/String.h"
#include <sstream>
int main(int argc, char **argv)
{
    ros::init(argc, argv, "talker");
    ros::NodeHandle n;
    ros::Publisher chatter_pub = n.advertise<std_msgs::String>("chatter", 1000);
    ros::Rate loop_rate(10);
    int count = 0;
    while (ros::ok())
    {
        std_msgs::String msg;
        std::stringstream ss;
        ss << "hello world " << count;
        msg.data = ss.str();
        ROS_INFO("%s", msg.data.c_str());
        chatter_pub.publish(msg);
        ros::spinOnce();
        loop_rate.sleep();
        ++count;
    }
    return 0;
}
```

Referência :
<http://wiki.ros.org/ROS/Tutorials/WritingPublisherSubscriber%28c%2B%2B%29>

Escrevendo um nodo que publica e um que se inscreve em um tópico

- Criando o nodo talker (pública):

Arquivo: **talker.cpp**

```
#include "ros/ros.h"
```

```
#include "std_msgs/String.h"
```

```
#include <sstream>
```

```
int main(int argc, char **argv)
```

```
{
```

```
  ros::init(argc, argv, "talker");
```

```
  ros::NodeHandle n;
```

```
  ros::Publisher chatter_pub = n.advertise<std_msgs::String>("chatter", 1000);
```

```
  ros::Rate loop_rate(10);
```

```
  int count = 0;
```

```
  while (ros::ok())
```

```
  {
```

```
    std_msgs::String msg;
```

```
    std::stringstream ss;
```

```
    ss << "hello world " << count;
```

```
    msg.data = ss.str();
```

```
    ROS_INFO("%s", msg.data.c_str());
```

```
    chatter_pub.publish(msg);
```

```
    ros::spinOnce();
```

```
    loop_rate.sleep();
```

```
    ++count;
```

```
  }
```

```
  return 0;
```

```
}
```

Include da biblioteca
cliente do ROS (roscpp)

Referência :

<http://wiki.ros.org/ROS/Tutorials/WritingPublisherSubscriber%28c%2B%2B%29>

Escrevendo um nodo que publica e um que se inscreve em um tópico

- Criando o nodo talker (pública):

Arquivo: **talker.cpp**

```
#include "ros/ros.h"
#include "std_msgs/String.h"
#include <sstream>
int main(int argc, char **argv)
{
    ros::init(argc, argv, "talker");
    ros::NodeHandle n;
    ros::Publisher chatter_pub = n.advertise<std_msgs::String>("chatter", 1000);
    ros::Rate loop_rate(10);
    int count = 0;
    while (ros::ok())
    {
        std_msgs::String msg;
        std::stringstream ss;
        ss << "hello world " << count;
        msg.data = ss.str();
        ROS_INFO("%s", msg.data.c_str());
        chatter_pub.publish(msg);
        ros::spinOnce();
        loop_rate.sleep();
        ++count;
    }
    return 0;
}
```

Include do tipo de
mensagem do tópico

Escrevendo um nodo que publica e um que se inscreve em um tópico

- Criando o nodo talker (pública):

Arquivo: **talker.cpp**

```
#include "ros/ros.h"
#include "std_msgs/String.h"
#include <sstream>
int main(int argc, char **argv)
{
    ros::init(argc, argv, "talker");
    ros::NodeHandle n;
    ros::Publisher chatter_pub = n.advertise<std_msgs::String>("chatter", 1000);
    ros::Rate loop_rate(10);
    int count = 0;
    while (ros::ok())
    {
        std_msgs::String msg;
        std::stringstream ss;
        ss << "hello world " << count;
        msg.data = ss.str();
        ROS_INFO("%s", msg.data.c_str());
        chatter_pub.publish(msg);
        ros::spinOnce();
        loop_rate.sleep();
        ++count;
    }
    return 0;
}
```

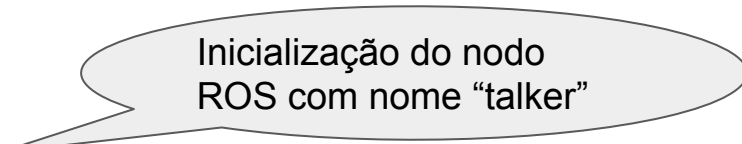
Include da biblioteca padrão do C++ para manipulação de stream (fluxo de texto)

Escrevendo um nodo que publica e um que se inscreve em um tópico

- Criando o nodo talker (pública):

Arquivo: **talker.cpp**

```
#include "ros/ros.h"
#include "std_msgs/String.h"
#include <sstream>
int main(int argc, char **argv)
{
    ros::init(argc, argv, "talker");
    ros::NodeHandle n;
    ros::Publisher chatter_pub = n.advertise<std_msgs::String>("chatter", 1000);
    ros::Rate loop_rate(10);
    int count = 0;
    while (ros::ok())
    {
        std_msgs::String msg;
        std::stringstream ss;
        ss << "hello world " << count;
        msg.data = ss.str();
        ROS_INFO("%s", msg.data.c_str());
        chatter_pub.publish(msg);
        ros::spinOnce();
        loop_rate.sleep();
        ++count;
    }
    return 0;
}
```



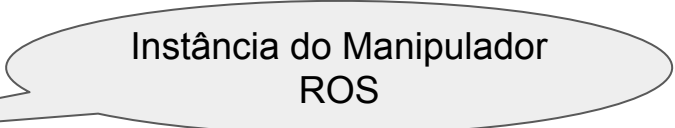
Inicialização do nodo
ROS com nome “talker”

Escrevendo um nodo que publica e um que se inscreve em um tópico

- Criando o nodo talker (pública):

Arquivo: **talker.cpp**

```
#include "ros/ros.h"
#include "std_msgs/String.h"
#include <sstream>
int main(int argc, char **argv)
{
    ros::init(argc, argv, "talker");
    ros::NodeHandle n;
    ros::Publisher chatter_pub = n.advertise<std_msgs::String>("chatter", 1000);
    ros::Rate loop_rate(10);
    int count = 0;
    while (ros::ok())
    {
        std_msgs::String msg;
        std::stringstream ss;
        ss << "hello world " << count;
        msg.data = ss.str();
        ROS_INFO("%s", msg.data.c_str());
        chatter_pub.publish(msg);
        ros::spinOnce();
        loop_rate.sleep();
        ++count;
    }
    return 0;
}
```



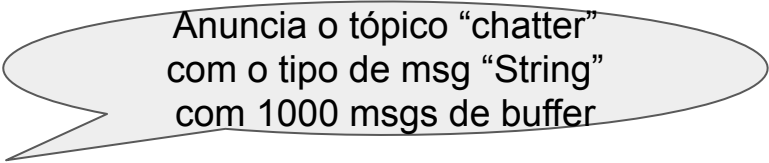
Instância do Manipulador
ROS

Escrevendo um nodo que publica e um que se inscreve em um tópico

- Criando o nodo talker (pública):

Arquivo: **talker.cpp**

```
#include "ros/ros.h"
#include "std_msgs/String.h"
#include <sstream>
int main(int argc, char **argv)
{
    ros::init(argc, argv, "talker");
    ros::NodeHandle n;
    ros::Publisher chatter_pub = n.advertise<std_msgs::String>("chatter", 1000);
    ros::Rate loop_rate(10);
    int count = 0;
    while (ros::ok())
    {
        std_msgs::String msg;
        std::stringstream ss;
        ss << "hello world " << count;
        msg.data = ss.str();
        ROS_INFO("%s", msg.data.c_str());
        chatter_pub.publish(msg);
        ros::spinOnce();
        loop_rate.sleep();
        ++count;
    }
    return 0;
}
```



Anuncia o tópico “chatter”
com o tipo de msg “String”
com 1000 msgs de buffer

Escrevendo um nodo que publica e um que se inscreve em um tópico

- Criando o nodo talker (pública):

Arquivo: **talker.cpp**

```
#include "ros/ros.h"
#include "std_msgs/String.h"
#include <sstream>
int main(int argc, char **argv)
{
    ros::init(argc, argv, "talker");
    ros::NodeHandle n;
    ros::Publisher chatter_pub = n.advertise<std_msgs::String>("chatter", 1000);
    ros::Rate loop_rate(10);
    int count = 0;
    while (ros::ok())
    {
        std_msgs::String msg;
        std::stringstream ss;
        ss << "hello world " << count;
        msg.data = ss.str();
        ROS_INFO("%s", msg.data.c_str());
        chatter_pub.publish(msg);
        ros::spinOnce();
        loop_rate.sleep();
        ++count;
    }
    return 0;
}
```

Instância um Rate de 10 Hz

Pausa de 1/10 segundos

Escrevendo um nodo que publica e um que se inscreve em um tópico

- Criando o nodo talker (pública):

Arquivo: **talker.cpp**

```
#include "ros/ros.h"
#include "std_msgs/String.h"
#include <sstream>
int main(int argc, char **argv)
{
    ros::init(argc, argv, "talker");
    ros::NodeHandle n;
    ros::Publisher chatter_pub = n.advertise<std_msgs::String>("chatter", 1000);
    ros::Rate loop_rate(10);
    int count = 0;
    while (ros::ok())
    {
        std_msgs::String msg;
        std::stringstream ss;
        ss << "hello world " << count;
        msg.data = ss.str();
        ROS_INFO("%s", msg.data.c_str());
        chatter_pub.publish(msg);
        ros::spinOnce();
        loop_rate.sleep();
        ++count;
    }
    return 0;
}
```

Cria um contador de msgs
utilizado para diferenciar msgs
transmitidas

Escrevendo um nodo que publica e um que se inscreve em um tópico

- Criando o nodo talker (pública):

Arquivo: **talker.cpp**

```
#include "ros/ros.h"
#include "std_msgs/String.h"
#include <sstream>
int main(int argc, char **argv)
{
    ros::init(argc, argv, "talker");
    ros::NodeHandle n;
    ros::Publisher chatter_pub = n.advertise<std_msgs::String>("chatter", 1000);
    ros::Rate loop_rate(10);
    int count = 0;
    while (ros::ok())
    {
        std_msgs::String msg;
        std::stringstream ss;
        ss << "hello world " << count;
        msg.data = ss.str();
        ROS_INFO("%s", msg.data.c_str());
        chatter_pub.publish(msg);
        ros::spinOnce();
        loop_rate.sleep();
        ++count;
    }
    return 0;
}
```

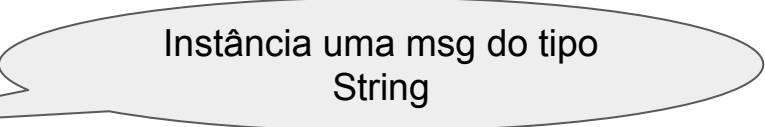
Repete enquanto existir
conexão.

Escrevendo um nodo que publica e um que se inscreve em um tópico

- Criando o nodo talker (pública):

Arquivo: **talker.cpp**

```
#include "ros/ros.h"
#include "std_msgs/String.h"
#include <sstream>
int main(int argc, char **argv)
{
    ros::init(argc, argv, "talker");
    ros::NodeHandle n;
    ros::Publisher chatter_pub = n.advertise<std_msgs::String>("chatter", 1000);
    ros::Rate loop_rate(10);
    int count = 0;
    while (ros::ok())
    {
        std_msgs::String msg;
        std::stringstream ss;
        ss << "hello world " << count;
        msg.data = ss.str();
        ROS_INFO("%s", msg.data.c_str());
        chatter_pub.publish(msg);
        ros::spinOnce();
        loop_rate.sleep();
        ++count;
    }
    return 0;
}
```



Instância uma msg do tipo
String

Escrevendo um nodo que publica e um que se inscreve em um tópico

- Criando o nodo talker (pública):

Arquivo: **talker.cpp**

```
#include "ros/ros.h"
#include "std_msgs/String.h"
#include <sstream>
int main(int argc, char **argv)
{
    ros::init(argc, argv, "talker");
    ros::NodeHandle n;
    ros::Publisher chatter_pub = n.advertise<std_msgs::String>("chatter", 1000);
    ros::Rate loop_rate(10);
    int count = 0;
    while (ros::ok())
    {
        std_msgs::String msg;
        std::stringstream ss;
        ss << "hello world " << count;
        msg.data = ss.str();
        ROS_INFO("%s", msg.data.c_str());
        chatter_pub.publish(msg);
        ros::spinOnce();
        loop_rate.sleep();
        ++count;
    }
    return 0;
}
```

Instância uma stream de texto

Escrevendo um nodo que publica e um que se inscreve em um tópico

- Criando o nodo talker (pública):

Arquivo: **talker.cpp**

```
#include "ros/ros.h"
#include "std_msgs/String.h"
#include <sstream>
int main(int argc, char **argv)
{
    ros::init(argc, argv, "talker");
    ros::NodeHandle n;
    ros::Publisher chatter_pub = n.advertise<std_msgs::String>("chatter", 1000);
    ros::Rate loop_rate(10);
    int count = 0;
    while (ros::ok())
    {
        std_msgs::String msg;
        std::stringstream ss;
        ss << "hello world " << count;
        msg.data = ss.str();
        ROS_INFO("%s", msg.data.c_str());
        chatter_pub.publish(msg);
        ros::spinOnce();
        loop_rate.sleep();
        ++count;
    }
    return 0;
}
```

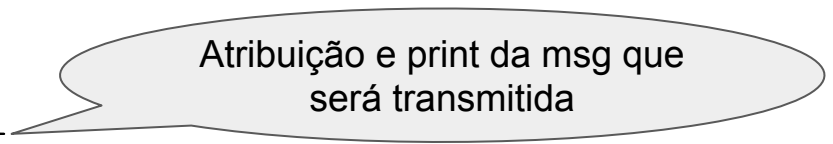
Constrói a msg que será transmitida incluindo o id da msg.

Escrevendo um nodo que publica e um que se inscreve em um tópico

- Criando o nodo talker (pública):

Arquivo: **talker.cpp**

```
#include "ros/ros.h"
#include "std_msgs/String.h"
#include <sstream>
int main(int argc, char **argv)
{
    ros::init(argc, argv, "talker");
    ros::NodeHandle n;
    ros::Publisher chatter_pub = n.advertise<std_msgs::String>("chatter", 1000);
    ros::Rate loop_rate(10);
    int count = 0;
    while (ros::ok())
    {
        std_msgs::String msg;
        std::stringstream ss;
        ss << "hello world " << count;
        msg.data = ss.str();
        ROS_INFO("%s", msg.data.c_str());
        chatter_pub.publish(msg);
        ros::spinOnce();
        loop_rate.sleep();
        ++count;
    }
    return 0;
}
```



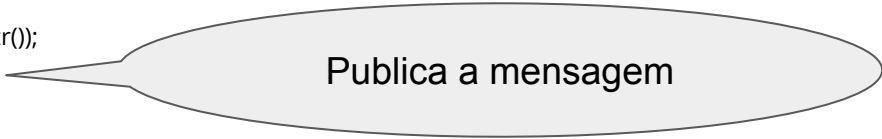
Atribuição e print da msg que
será transmitida

Escrevendo um nodo que publica e um que se inscreve em um tópico

- Criando o nodo talker (pública):

Arquivo: **talker.cpp**

```
#include "ros/ros.h"
#include "std_msgs/String.h"
#include <sstream>
int main(int argc, char **argv)
{
    ros::init(argc, argv, "talker");
    ros::NodeHandle n;
    ros::Publisher chatter_pub = n.advertise<std_msgs::String>("chatter", 1000);
    ros::Rate loop_rate(10);
    int count = 0;
    while (ros::ok())
    {
        std_msgs::String msg;
        std::stringstream ss;
        ss << "hello world " << count;
        msg.data = ss.str();
        ROS_INFO("%s", msg.data.c_str());
        chatter_pub.publish(msg);
        ros::spinOnce();
        loop_rate.sleep();
        ++count;
    }
    return 0;
}
```



Publica a mensagem

Escrevendo um nodo que publica e um que se inscreve em um tópico

- Criando o nodo talker (pública):

Arquivo: **talker.cpp**

```
#include "ros/ros.h"
#include "std_msgs/String.h"
#include <sstream>
int main(int argc, char **argv)
{
    ros::init(argc, argv, "talker");
    ros::NodeHandle n;
    ros::Publisher chatter_pub = n.advertise<std_msgs::String>("chatter", 1000);
    ros::Rate loop_rate(10);
    int count = 0;
    while (ros::ok())
    {
        std_msgs::String msg;
        std::stringstream ss;
        ss << "hello world " << count;
        msg.data = ss.str();
        ROS_INFO("%s", msg.data.c_str());
        chatter_pub.publish(msg);
        ros::spinOnce();
        loop_rate.sleep();
        ++count;
    }
    return 0;
}
```

Processa todas requisições internas do ROS, como recepção de mensagens, chamada de callbacks, uma única vez.

Escrevendo um nodo que publica e um que se inscreve em um tópico

- Criando o nodo talker (pública):

Arquivo: **talker.cpp**

```
#include "ros/ros.h"
#include "std_msgs/String.h"
#include <sstream>
int main(int argc, char **argv)
{
    ros::init(argc, argv, "talker");
    ros::NodeHandle n;
    ros::Publisher chatter_pub = n.advertise<std_msgs::String>("chatter", 1000);
    ros::Rate loop_rate(10);
    int count = 0;
    while (ros::ok())
    {
        std_msgs::String msg;
        std::stringstream ss;
        ss << "hello world " << count;
        msg.data = ss.str();
        ROS_INFO("%s", msg.data.c_str());
        chatter_pub.publish(msg);
        ros::spinOnce();
        loop_rate.sleep();
        ++count;
    }
    return 0;
}
```

shorturl.at/agzJP

Referência :
<http://wiki.ros.org/ROS/Tutorials/WritingPublisherSubscriber%28c%2B%2B%29>