Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ПЕТРОЗАВОДСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
Институт математики и информационных технологий

Отчет о прохождении производственной практики

Выполнил студент(ка) группы 22207
Анисимов Никита Николаевич
(ФИО)

Направление подготовки:
Системное и прикладное программное обеспечение

Место прохождения практики:
Кафедра ИМО, ЦИИ

Сроки прохождения практики:
30.05.2022 – 09.06.2022

Руководитель практики:
к.т.н., доцент
Богоявленская Ольга Юрьевна

Оценка _____

Дата _____

Петрозаводск 2022 г.

Содержание

Введение

Цель практики: изучить механизм publisher/subscriber в системе ROS.
Задачи практики:

- изучить систему ROS
- изучить концепцию подписчика и издателя
- написать подписчика и издателя
- протестировать работу программ
- рассмотреть использование механизма в проектах

1 Система ROS

1.1 Теория

Я работал с системой ROS (Robot Operating System), которая является ОС для программирования роботов. Помимо реализации всех стандартных служб операционной системы, ROS предоставляет набор программ для удобной работы с пакетами, их сборки и тестирования. В частности систему catkin для сборки пакетов.

1.2 Практика

Я поставил VirtualBox с Ubuntu, на которую планировал поставить ROS Noetic. Далее поставил саму систему по инструкции с официального сайта, а так же настроил catkin workspace.

2 Publisher and subscriber

2.1 Теория

ROS основан на архитектуре графов, где обработка данных происходит в программах, называемых узлами (nodes), которые могут получать и передавать сообщения (messages) между собой. Одним из способов взаимодействия между ними в ROS является модель издатель-подписчик (publisher-subsriber). Ключевым объектом здесь является тема (topic), в которую издатель публикует сообщения, а подписчик считывает их оттуда. У одной темы может быть несколько как подписчиков, так и издателей. Тип сообщений можно использовать как стандартный, так и задать самому с помощью msg файлов.

2.2 Практика

Я создал пакет beginner_tutorials для моих программ (nodes). В манифест и CMakeLists добавил все необходимые завесимости. Так же создал msg файл Num.msg где описал формат сообщений. Далее написал две программы на языке C++:

- talker.cpp – издатель

- listener.cpp – подписчик

Обе программы взаимодействовали через тему chatter. Издатель публиковал значения увеличивающегося счетчика, а подписчик считывал их и выводил в консоль. Исходные коды находятся в приложении.

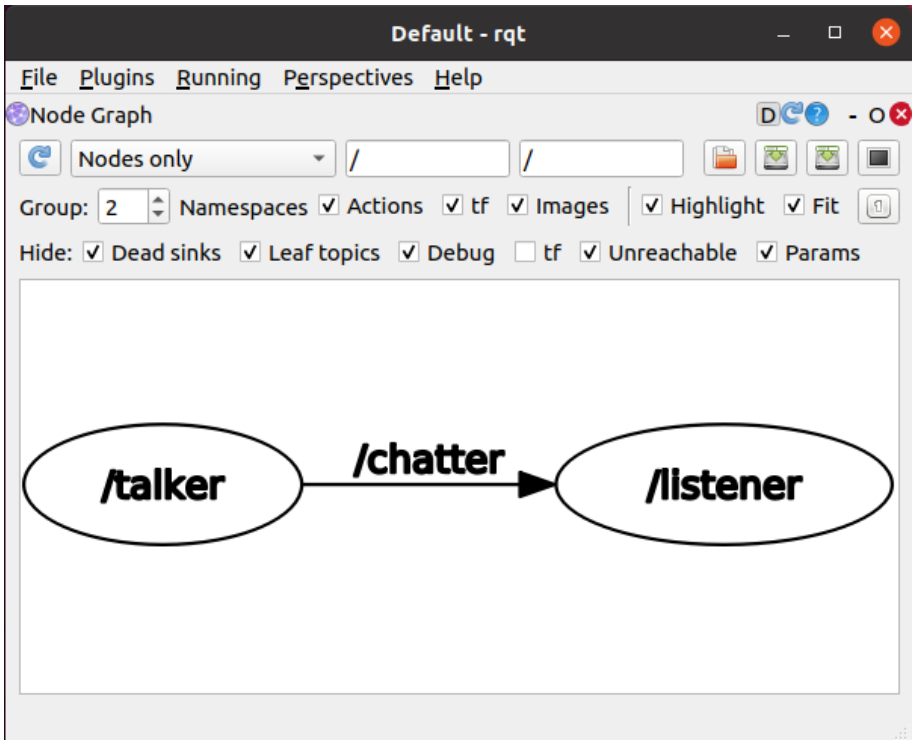

Рис. 1: Созданный с помощью rqt граф, изображающий происходящее в системе

```
nikita@nikita-VirtualBox:~/catkin_ws$ rosmsg show Num
[beginner_tutorials/Num]:
int64 num
```

Рис. 2: Тип сообщений Num

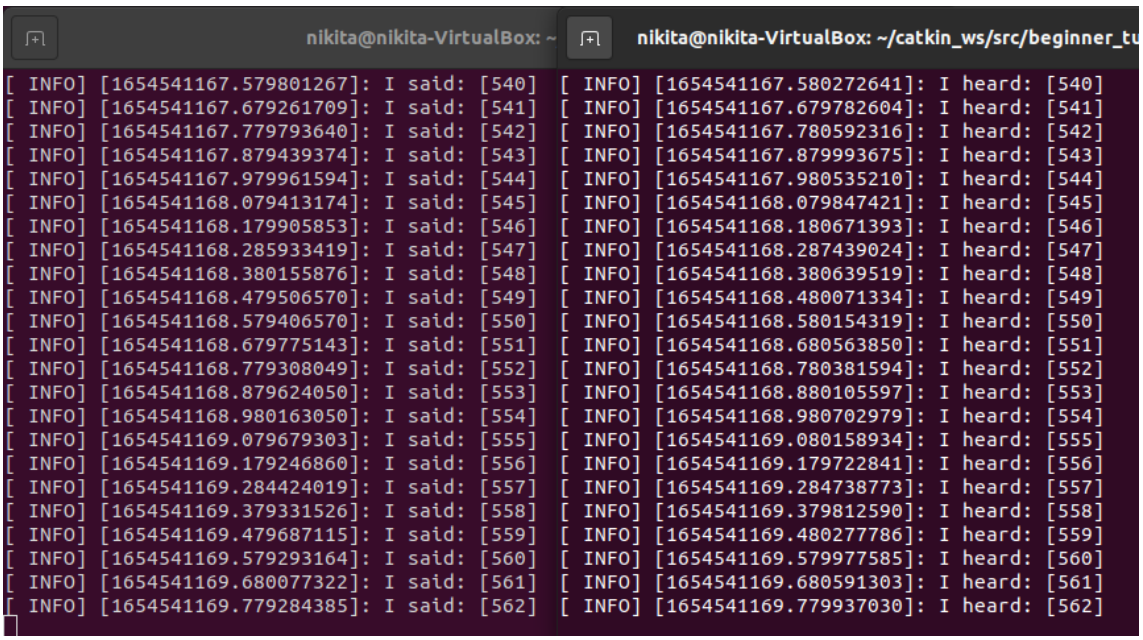

Рис. 3: Вывод программ

3 Использование в проектах

При разработки роботов концепция издатель-подписчик очень важна, т.к. обеспечивает обмен информацией между отдельными компонентами системы. Например узел А отвечает за датчики робота, и ему необходимо передовать данные о препятствиях на пути узлу Б, который отвечает за движение робота. Узел А является издателем и публикует сообщение об обстановке в тему "препятствие". Узел Б считывает оттуда сообщения и при уведомлении о препятствии останавливает движение робота и перестраивает маршрут.

Заключение

В данной практике была изучена система ROS и механизм publisher/subscriber, который в ней используется. Были написаны и протестированы программы, реализующие этот механизм, а также рассмотрен пример его использования в проекте робота.

Приложение А. Код издателя

```cpp
#include "ros/ros.h"
#include "beginner_tutorials/Num.h"

int main(int argc, char **argv)
{
  // Инициализация узла
  ros::init(argc, argv, "talker");

  // Интерфейс для создания узлов
  ros::NodeHandle n;

  // Создание издателя, который будет публиковать сообщения
  // в тему chatter в формате Num
  ros::Publisher chatter_pub =
    n.advertise<beginner_tutorials::Num>("chatter", 1000);

  // Частота цикла
  ros::Rate loop_rate(10);

  // Счетчик
  int count = 0;

  // Пока нет ошибок публикуем сообщения
  while (ros::ok())
  {
    // Создание сообщения
    beginner_tutorials::Num msg;
    msg.num = count;

    // Вывод в консоль
    ROS_INFO("I said: [%ld]", msg.num);

    // Публикация сообщения
    chatter_pub.publish(msg);

    // Задержка
    loop_rate.sleep();

    // Увеличение счетчика
    ++count;
  }
  return 0;
}
```

Приложение Б. Код подписчика

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

void chatterCallback(const beginner_tutorials::Num::ConstPtr& msg)
{
  // Консольный вывод
  ROS_INFO("I heard: [%ld]", msg->num);
}

int main(int argc, char **argv)
{
  // Инициализация узла
  ros::init(argc, argv, "listener");

  // Интерфейс для создания узлов
  ros::NodeHandle n;

  // Создание подписчика темы chatter, который при считывании сообщения
  // вызывает функцию chatterCallback с сообщением в параметре
  ros::Subscriber sub = n.subscribe("chatter", 1000, chatterCallback);

  // Вход в цикл считыванию сообщений
  ros::spin();
  return 0;
}
```