МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

“МИРЭА - Российский технологический университет”

РТУ МИРЭА

Институт кибернетики

Кафедра проблем управления

ЛАБОРАТОРНАЯ РАБОТА №4

по дисциплине

Программное обеспечение мехатронных и робототехнических систем

Тема работы: “Осуществление публикации и чтения данных внутри информационной среды ROS”

Студент группы КРБО-03-17

Орлов Д.Л.

Преподаватель: Морозов А.А._____

ЦЕЛЬ И ЗАДАНИЕ

Цель работы: Получение навыков установки и настройки ROS (Robot operating system), ознакомление с архитектурой ROS, создание узла, осуществляющего публикацию данных и чтение данных из информационной среды ROS. **Задание:** Установить и настроить ROS. Создать узлы Publisher и Subscriber для публикации и чтения данных из информационной среды ROS.

ХОД РАБОТЫ

Был выбран дистрибутив ROS Melodic. Для начала настроим систему, для того что бы мы могли принимать программное обеспечение, выполнив команду:

```
sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu
$(lsb_release -sc) main" > /etc/apt/sources.list.d/ros-
latest.list'
```

Настроим ключи доступа в системе для правильной загрузки:

```
sudo apt-key adv --keyserver hkp://ha.pool.sks-keyservers.net:80 -
    -recv-key          C1CF6E31E6BADE8868B172B4F42ED6FBAB17C654
```

Убедимся в том, что мы имеем последние версии индексов пакетов:

```
sudo apt-get update
```

Теперь установим сам пакет ROS. Так как мы решили использовать ROS вместе с симуляцией, установим полную версию.

```
sudo apt-get install ros-melodic-desktop-full
```

Далее необходимо инициализировать rosdep:

```
sudo rosdep init rosdep
update
```

Настроим переменное окружение так, чтобы оно добавлялось в наш сеанс bash при каждом запуске новой оболочки:

```
echo "source /opt/ros/melodic/setup.bash" >> ~/.bashrc source
~/.bashrc
```

Далее при помощи команды запустим ROS и добавим узел для симуляции черепашки, таким образом убедимся в правильности установки (рисунок 1).

```
roscore
rosrun turtlesim turtlesim_node
```

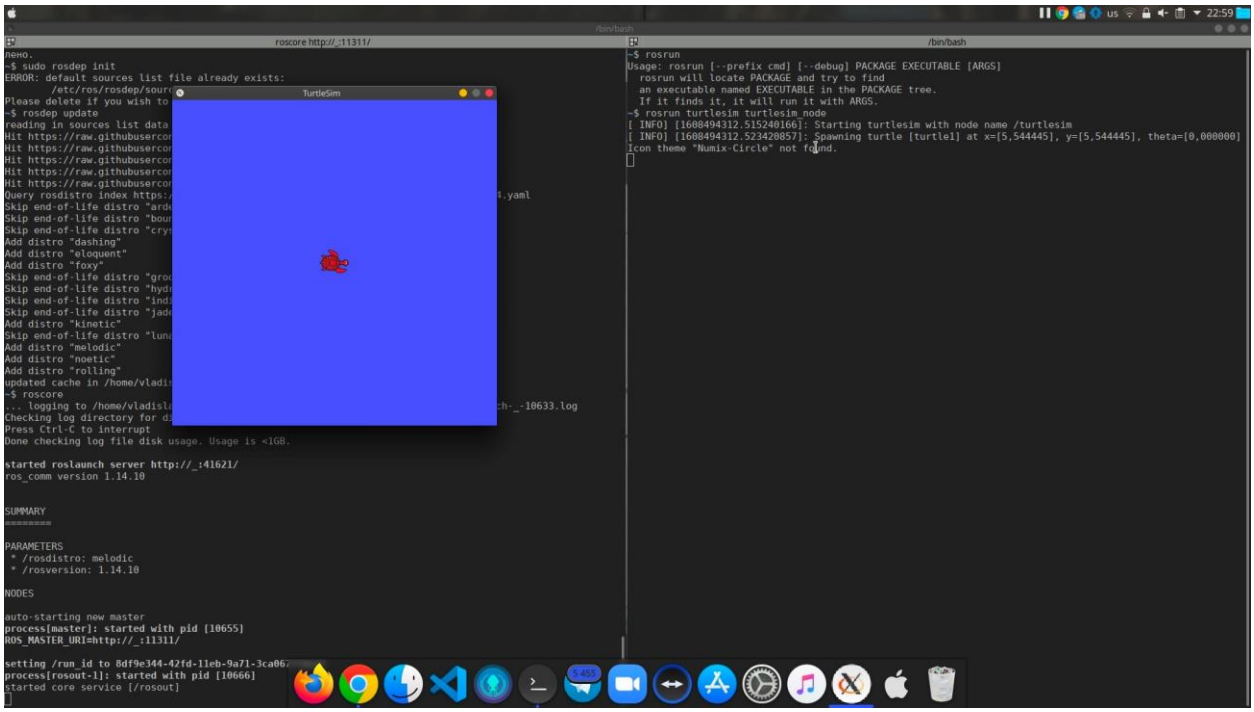

Рисунок 1 - Запуск ROS и turtlesim

Теперь необходимо создать рабочее пространство для создания своего пакета ROS.

Создадим папку рабочего пространства.

```
mkdir -p ~/ros_workspace/src
```

Проинициализируем рабочее пространство и соберем пакет.

```
cd ~/ros_workspace && catkin_init_workspace && catkin_make
```

Создадим новый пакет OrlovDL.

```
cd ~/ros_workspace/src && catkin_create_pkg OrlovDL std_msgs rospy
```

Создадим файлы скриптов Publisher и Subscriber. Сделаем их исполняемыми.

```
mkdir ~/ros_workspace/src/scripts
cd ~/ros_workspace/src/scripts
touch OrlovDL_publisher.py OrlovDL_subscriber.py
chmod +x OrlovDL_publisher.py OrlovDL_subscriber.py
```

Напишем код скриптов (см. приложение) и пересоберем пакет.

```
cd ~/ros_workspace && catkin_init_workspace && catkin_make
```

Запустим ROS, ноду-публикатора и ноду-слушателя в трёх терминалах, таким образом проверим работу (рисунок 2).

```
roscore
rosrun OrlovDL OrlovDL_publisher.py
rosrun OrlovDL OrlovDL_subscriber.py
```

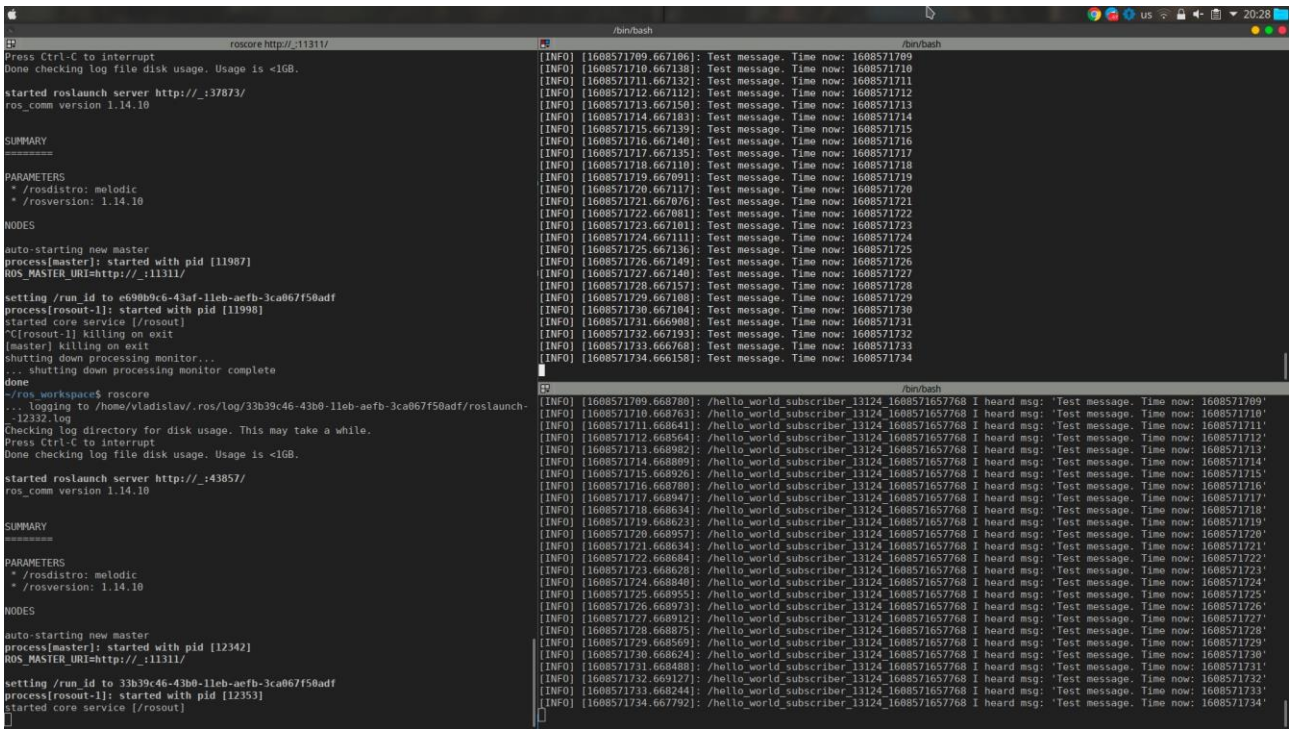

Рисунок 2 - Демонстрация работы Publisher и Subscriber

ВЫВОД

В ходе лабораторной работы были получены практические навыки по установке и настройке среды ROS-Melodic. Была настроена рабочая среда для создания новых проектов. Был создан и собран новый проект “ `OrlovDL`”, который реализует публикацию и чтение данных в информационную среду ROS.

ПРИЛОЖЕНИЕ

OrlovDL_publisher

```
#!/usr/bin/env python import
rospy from std_msgs.msg import
String

def talker():
    Pub = rospy.Publisher('hello_pub', String, queue_size=10)
rospy.init_node(OrlovDL_publisher', anonymous=True)     r =
rospy.Rate(1)

    while not rospy.is_shutdown():
        msg = "Test message. Time now: " + str(rospy.get_time())
[:-3]
rospy.loginfo(msg)
OrlovDL.publish(msg)
        r.sleep()

if __name__ == '__main__':
try:
        talker()
    except rospy.ROSInterruptException: pass
```

OrlovDL_subscriber

```
#!/usr/bin/env python import
rospy from std_msgs.msg import
String

def callback(data):
    rospy.loginfo(rospy.get_caller_id() + " I heard msg: '%s'",
data.data)

def listener():
```

```
    rospy.init_node(' OrlovDL_subscriber', anonymous=True)
rospy.Subscriber("hello_pub", String, callback)     rospy.spin()

if __name__ == '__main__':
    listener()
```