**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра математического обеспечения и применения ЭВМ**

отчет

**по лабораторной работе №1**

**по дисциплине «Системы реального временина основе Linux»**

|  |  |  |
| --- | --- | --- |
| Студент гр. 2303 |  | Маняшин Д.А. |
| Преподаватель |  | Филатов А.Ю. |

Санкт-Петербург

2017

**Цель работы.**

Необходимо при помощи Robot Operation System(далее РОС) реализовать модель хождения объектов в лабиринте.

**Выполнение работы**

Было принято решения остановится на языке программирования Python ввиду его синтаксической простоты.

1. Создание ноды слушателя и подписываем ее на топик lab\_1. В ходе работы пришлось реализовать класс робот содержавший в себе информацию о машине, основные методы для перемещения в пространстве при запуске ноды производится генерация лабиринта и робота в пространстве. Код слушателя приведен в приложении А
2. Создание ноды отправителя и подписывание ее на токи. Для этой ноды прилошлось реализовать перехват ручного управления роботом при помощи клавиш клавиатуры для этого была выбрана библиотека pyinput. В результате при нажатии определенных клавиш про исходит отправка сообщение в топик lab\_1, где их в дальнейшем обрабатывает слушатель. Всего в работе реализованно 6 команд; влево, вправо, вниз, вверх, показать карту и выйти из приложения.

**Диаграмма**

Отобразим диаграмму на нод и топиков(рисунок 1)

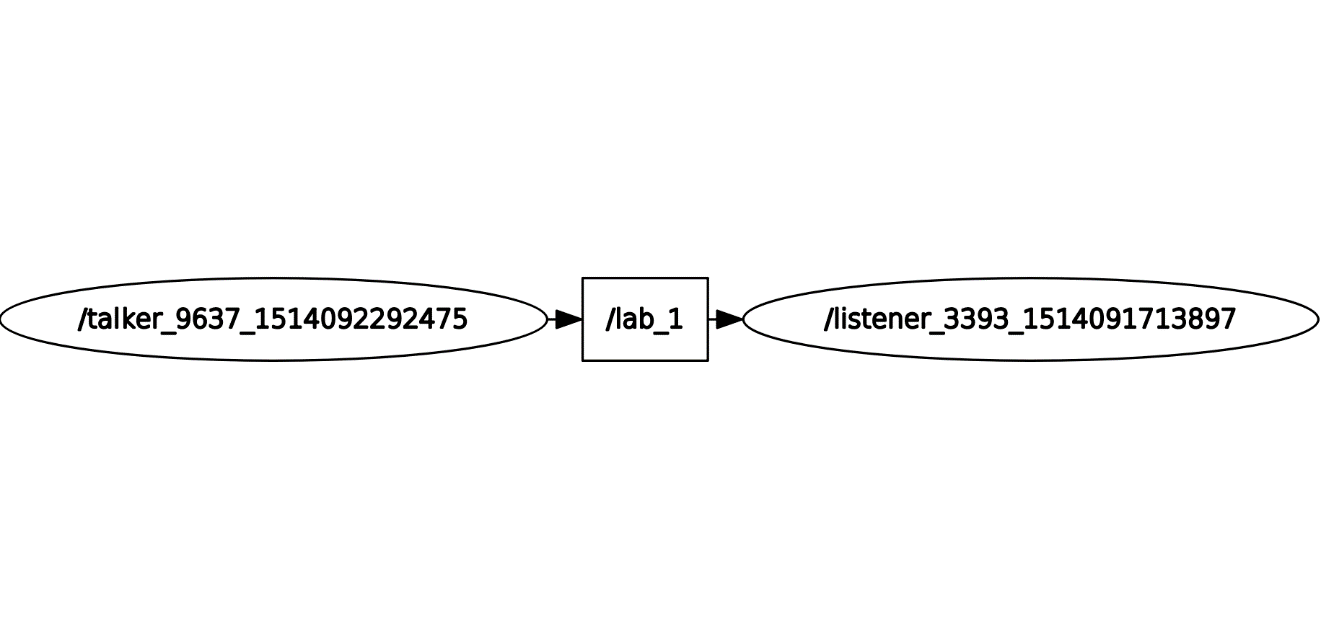
****

Рисунок 1 диаграмма нод и топиков

**Проверка результатов**

Для проверки резутатов кода запустим основное приложение roscore в отдельном терминале так-как все ноды требует отдельный терминал придется открыть еще 2 терминала для слушателя и отправителя сообщений. На рисунке представлены термина с выполняемым приложением

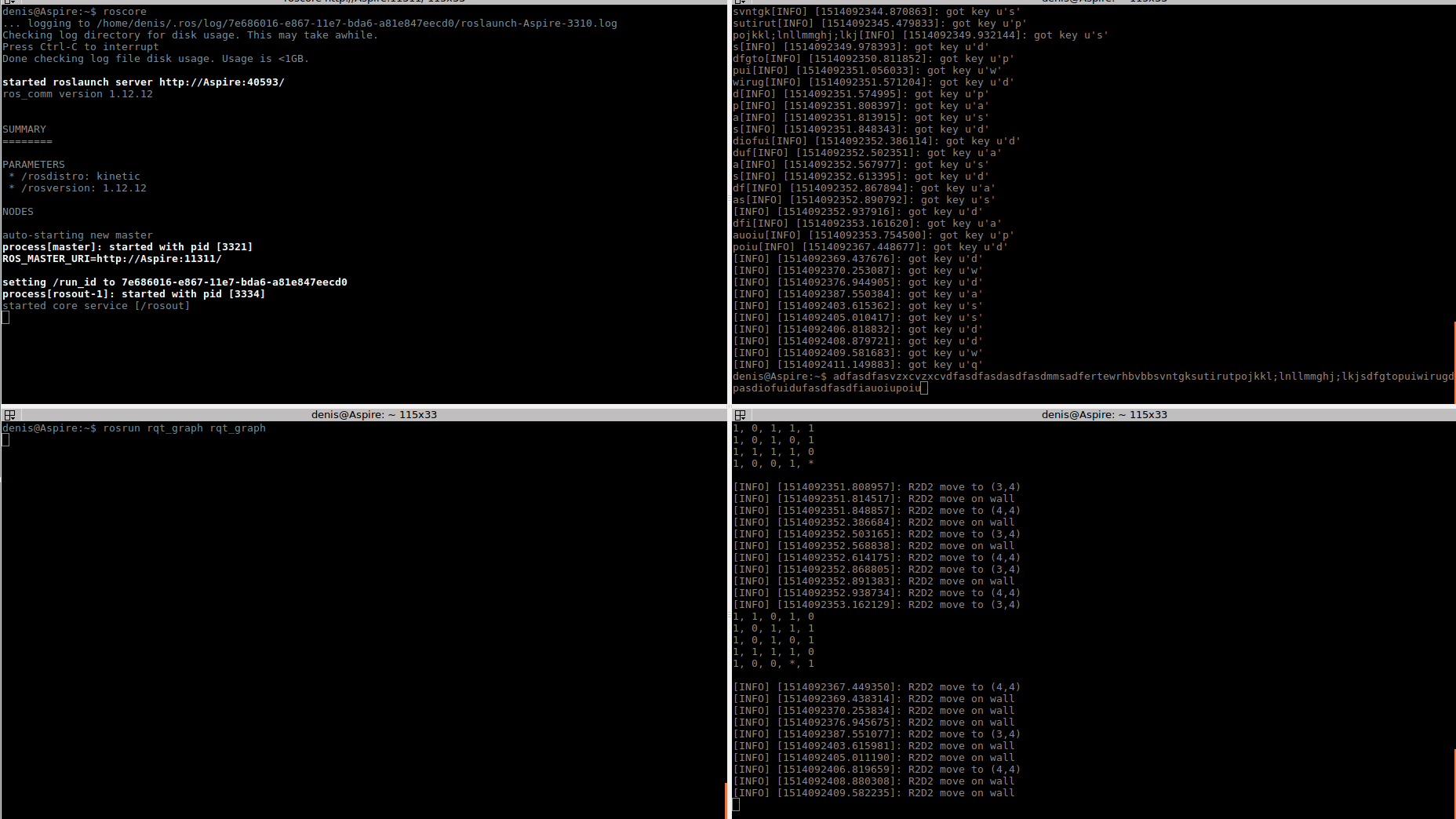


Рисунок 2 Терминалы с проложением

**Выводы.**

В ходе лабораторной работы были освоены основные компоненты Рос топики и ноды, путем создания модели робота в лабиринте.

**Приложение А**

**Код слушателя**

#!/usr/bin/env python

import rospy

from std\_msgs.msg import String

import random

import copy

topik = "lab\_1"

class Robot:

def \_\_init\_\_(self,map = None, name="R2D2", x=2, y=2):

self.name = name

self.x = x

self.y = y

self.map = map

def move(self, dx, dy):

new\_x = self.x + dx

new\_y = self.y + dy

if self.is\_free(new\_x, new\_y):

rospy.loginfo('%s move to (%d,%d)', self.name,new\_x, new\_y)

self.x = new\_x

self.y = new\_y

else:

rospy.loginfo('%s move on wall' % self.name)

def show\_location(self):

map = copy.deepcopy(self.map)

map[self.y][self.x] = "\*"

for i in map:

print ', '.join(str(v) for v in i)

print ""

del map

def is\_free(self,x,y):

if (x >= 0 and x < len(self.map[0]) and y >= 0 and y < len(self.map)):

return self.map[y][x] == 1

else:

return False

def run\_on(self, topic):

def run\_callback(data):

if data.data == 'up':

self.move(0, -1)

if data.data == 'left':

self.move(-1, 0)

if data.data == 'right':

self.move(1, 0)

if data.data == 'down':

self.move(0, 1)

if data.data == 'print':

self.show\_location()

rospy.init\_node('listener', anonymous=True)

rospy.Subscriber(topic, String, run\_callback)

rospy.spin()

def callback(data):

rospy.loginfo('%s - get data %s' % (rospy.get\_time(),data.data))

def genarate\_map(mx=5,my=5):

nmap = []

imgx = mx \* 1

imgy = my \* 1

maze = [[0 for x in range(mx)] for y in range(my)]

dx = [0, 1, 0, -1]

dy = [-1, 0, 1, 0]

stack = [(random.randint(0, mx - 1), random.randint(0, my - 1))]

while len(stack) > 0:

(cx, cy) = stack[-1]

maze[cy][cx] = 1

nlst = []

for i in range(4):

nx = cx + dx[i]

ny = cy + dy[i]

if nx >= 0 and nx < mx and ny >= 0 and ny < my:

if maze[ny][nx] == 0:

ctr = 0

for j in range(4):

ex = nx + dx[j];

ey = ny + dy[j]

if ex >= 0 and ex < mx and ey >= 0 and ey < my:

if maze[ey][ex] == 1: ctr += 1

if ctr == 1: nlst.append(i)

if len(nlst) > 0:

ir = nlst[random.randint(0, len(nlst) - 1)]

cx += dx[ir];

cy += dy[ir]

stack.append((cx, cy))

else:

stack.pop()

for ky in range(imgy):

line = []

for kx in range(imgx):

value = maze[my \* ky / imgy][mx \* kx / imgx]

line.append(value)

nmap.append(line)

return nmap

def listener():

mrobot = Robot(genarate\_map())

mrobot.run\_on(topik)

if \_\_name\_\_ == '\_\_main\_\_':

listener()

**Приложение Б**

#!/usr/bin/env python

import rospy

from std\_msgs.msg import String

import random

from pynput.keyboard import Key, Listener, KeyCode

def generate\_key():

key = random.randint(1,6)

if key == 1: return "up"

if key == 2: return "down"

if key == 3: return "left"

if key == 4: return "right"

if key == 5: return "print"

topik = "lab\_1"

\_Q\_ = KeyCode(char='q')

\_W\_ = KeyCode(char='w')

\_A\_ = KeyCode(char='a')

\_S\_ = KeyCode(char='s')

\_D\_ = KeyCode(char='d')

\_P\_ = KeyCode(char='p')

keys = (\_Q\_, \_W\_, \_A\_, \_S\_, \_D\_, \_P\_, Key.esc)

pub = rospy.Publisher(topik, String, queue\_size=1000)

def on\_press(key):

if key in keys:

rospy.loginfo('got key %s', key)

if key == \_W\_:

pub.publish('up')

if key == \_S\_:

pub.publish('down')

if key == \_A\_:

pub.publish('left')

if key == \_D\_:

pub.publish('right')

if key == \_P\_:

pub.publish('print')

def on\_release(key):

if key in (Key.esc, \_Q\_):

return False

def talker():

rospy.init\_node('talker', anonymous=True)

with Listener(on\_press=on\_press, on\_release=on\_release

) as listener:

listener.join(

if \_\_name\_\_ == '\_\_main\_\_':

try:

talker()

except rospy.ROSInterruptException:

pass