

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра МО ЭВМ**

**ОТЧЕТ**  
**по лабораторной работе №2**  
**по дисциплине «ROS (Операционная система роботов)»**  
**Тема: Понятие сервиса в ROS. Обмен запросами в ROS.**

Студент гр. 2304

Мотовилов И.А.

Преподаватель

Филатов А.Ю.

Санкт-Петербург

2017

## **Цель работы.**

Изучение концепции клиента-сервера в ROS, а также создание игрового пакета, реализующего данный шаблон.

## **Основные теоретические положения.**

ROS - это фреймворк для программирования роботов, предоставляющий функциональность для распределённой работы.

Узлы (Nodes): узел — это исполняемый файл, который использует ROS для связи с другими узлами.

Сервисы (Service): узлы могут отправлять запрос другим узлам, в свою очередь те, могут отправить обратно ответ.

В отличие от концепции издатель-подписчик, концепция клиент-сервер подразумевает под собой то, что клиент отправляет запрос с определенными параметрами на сервер, а сервер в свою очередь по запросу, выполняет какого-то действие, а так же может отправить клиенту в качестве ответа определенные параметры.

В рамках данной работы, был разработан игровой пакет ремонта самолета, который реализует концепцию клиента-сервера, а именно:

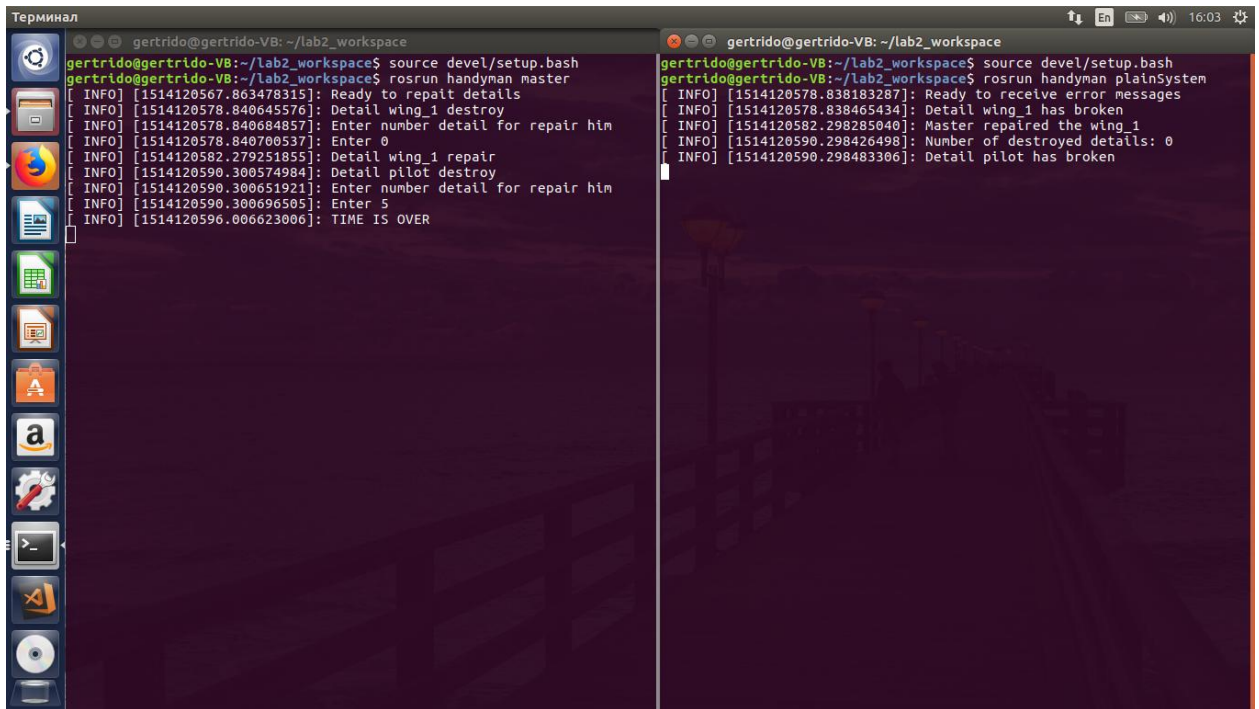
1. Узел "PlainSystem": данный узел выступает в качестве клиента. Каждые 8 секунд, случайным образом ломается какая-то деталь самолета, как только узел это замечает, то отправляет запрос на сервер, где в качестве параметра выступает код сломанной детали.
2. Узел "Master": данный узел выступает в качестве сервера сервиса. Как только придет запрос, то узел предлагает в течение 5 секунд ввести код сломанной детали. Если данная операция проходит успешно, то сервер "чинит деталь", т.е. отправляет в качестве ответа "true", иначе отправляет false.
3. Точкой выхода из программы является ситуация, когда количество сломанных деталей стало больше 3.

## Экспериментальные результаты.

Для демонстрации результатов, необходимо:

1. Запустить сервер ROS: roscore
2. Запустить узел мастера: rosrund handyman master
3. Запустить узел системы самолета: rosrund handyman plainSystem

Результат работ представлен на рис.1



The image shows two terminal windows side-by-side, both titled 'Терминал' and 'gertrido@gertrido-VB: ~/lab2\_workspace'. The left terminal shows the execution of 'roslaunch handyman master' and 'roslaunch handyman plainSystem'. It displays a series of log messages including 'Ready to repair details', 'Detail wing\_1 destroy', 'Enter number detail for repair him', 'Detail wing\_1 repair', 'Detail pilot destroy', and 'TIME IS OVER'. The right terminal shows the execution of 'roslaunch handyman plainSystem' and displays log messages including 'Ready to receive error messages', 'Detail wing\_1 has broken', 'Master repaired the wing\_1', 'Number of destroyed details: 0', and 'Detail pilot has broken'. The background of the terminal windows is a dark purple with a faint grid pattern.

рис.1 - демонстрация работы пакета

## Выводы.

В результате выполнения работы были изучены понятия сервиса и клиент-серверная концепция в ROS, а также был успешно реализован игровой пакет, демонстрирующий работу сервиса в ROS.