

Моделирование колесных роботов

Семинар 3. Сервисы и действия,
параметры, roslaunch

Андрей Соколов

Москва, 2022

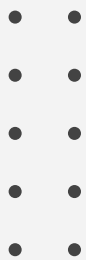


ИППИ РАН

NATIONAL RESEARCH
UNIVERSITY



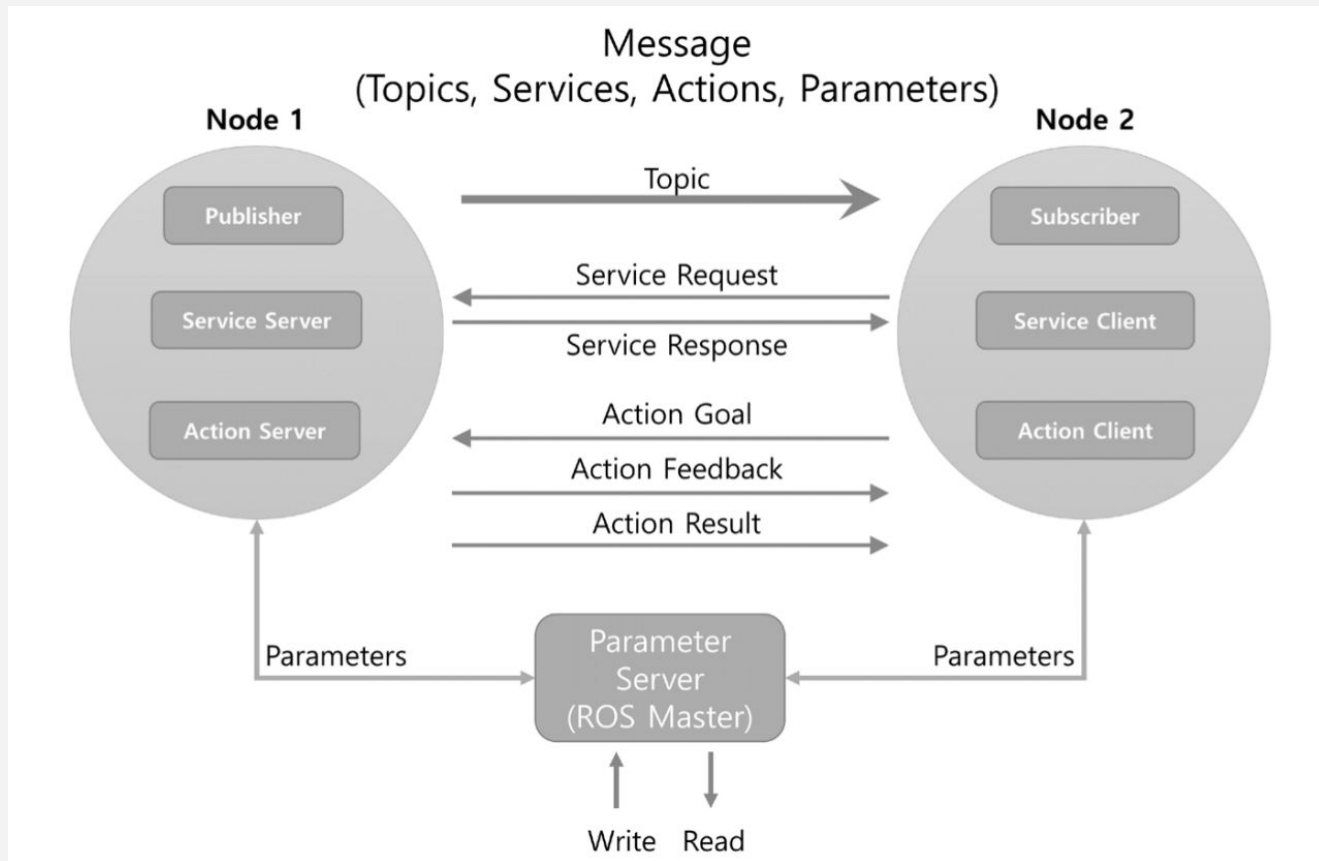
<https://qrgo.page.link/SCcy1>



СОДЕРЖАНИЕ ЛЕКЦИИ

1. Сервисы
2. Действия
3. Сервер параметров
4. roslaunch

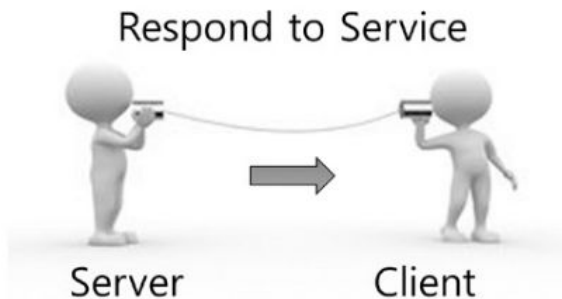
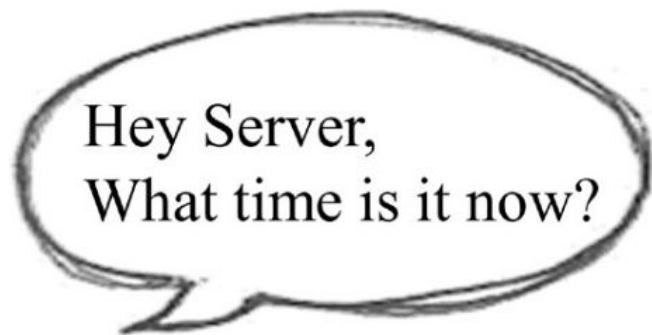
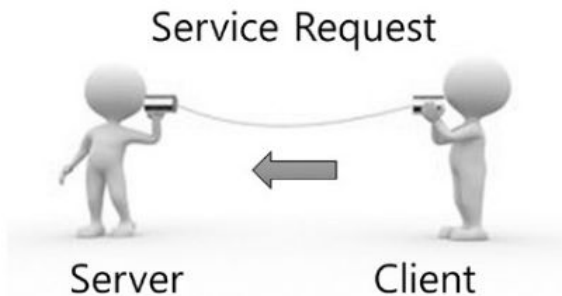
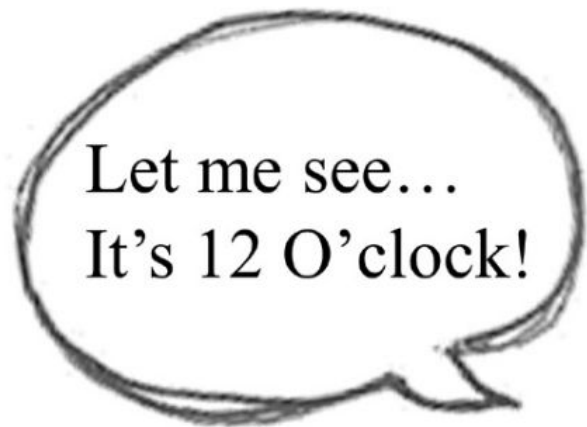
ТИПЫ КОММУНИКАЦИИ В ROS



ТИПЫ КОММУНИКАЦИИ В ROS

Тип коммуникации	Особенности	Когда применяется?
Топик	Асинхронная однонаправленная передача	Используется для непрерывной передачи данных
Сервис	Синхронная двунаправленная передача	Используется для коммуникации типа запрос-ответ с быстрой обработкой запроса
Действие	Асинхронная двунаправленная передача	Используется, когда механизм сервисов неприменим, из-за долгого процесса обработки запроса или когда необходима обратная связь в процессе обработки запроса

СЕРВИСЫ (SERVICES)



СЕРВИСЫ (SERVICES)

<http://wiki.ros.org/rospy/Overview/Services>

- ❑ Пакет `std_srvs` содержит набор стандартных сервисов
- ❑ Определение сервиса состоит из двух частей:
 - ❑ Запроса
 - ❑ Ответа
- ❑ Допускаются пустой запрос и пустой ответ
- ❑ **Запрос и ответ** могут иметь любой тип сообщения:
 - ❑ встроенный тип (например, `float64`)
 - ❑ тип уже существующего сообщения (`geometry_msgs/Quaternion`)
 - ❑ массив фиксированной или динамической длины (`float64[]` или `float64[9]`)

```
std_srvs/SetBool
```

```
bool data
# e.g. for hardware enabling /
disabling
---
bool success
# indicate successful run of
triggered service
string message
# informational, e.g. for error
messages
```

```
std_srvs/Empty
```

```
---
```

НАПИСАНИЕ СЕРВИС-СЕРВЕРА

- ❑ Импорт сервиса и ответа сервиса из пакета

```
from test_package.srv import GetWindowMedian, GetWindowMedianResponse
from <package>.srv import <Service>, <Service>Response
```

- ❑ Создание сервис-сервера

```
rospy.Service("get_median", GetWindowMedian, handle_get_median)
rospy.Service(name, service_class, handler, buff_size=65536, error_handler=None)
```

- ❑ Объявление обработчика запросов сервиса

```
def handle_get_median(req):
    # some service-handling code
    return GetWindowMedianResponse(<response_data>)
```


НАПИСАНИЕ СЕРВИС-КЛИЕНТА

- ❑ Блокирование выполнения до появления сервиса в системе

```
rospy.wait_for_service("get_median")  
rospy.wait_for_service(service, timeout=None)
```

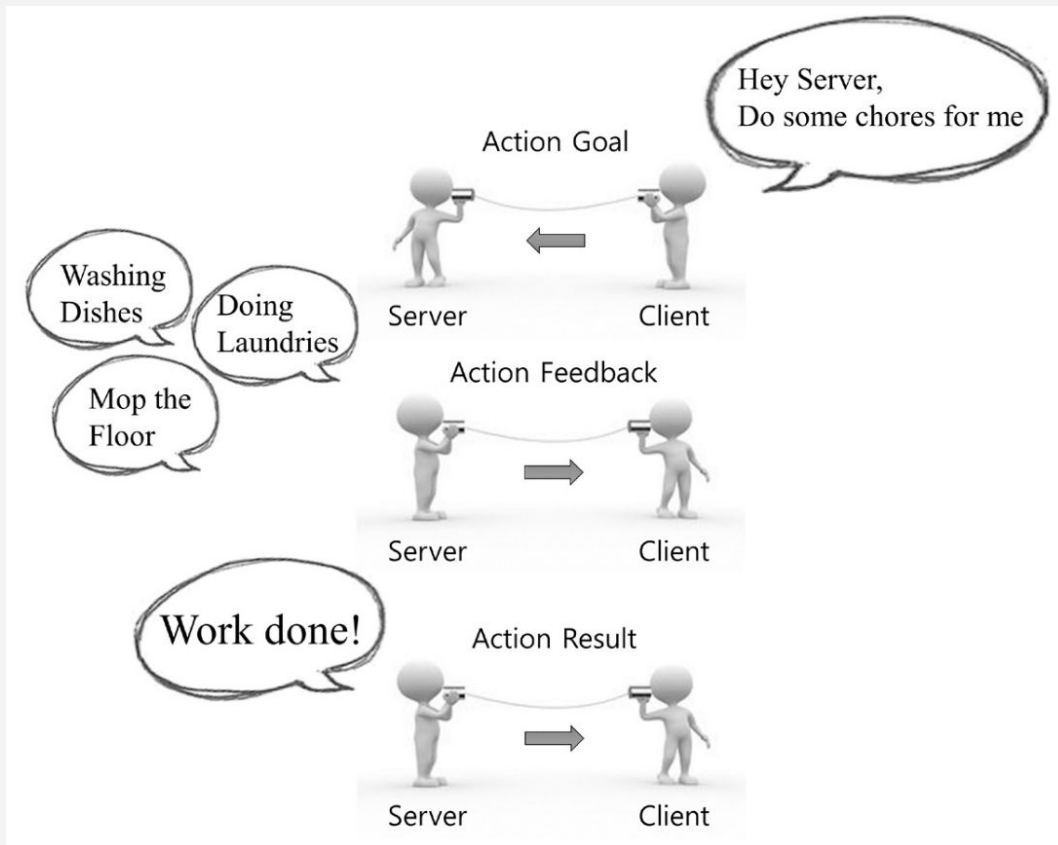
- ❑ Создание сервис-клиента

```
get_median = rospy.ServiceProxy("get_median", GetWindowMedian)  
rospy.ServiceProxy(name, service_class, persistent=False, headers=None)
```

- ❑ Вызов сервиса и получение ответа от сервера

```
response = get_median(<request_data>)
```

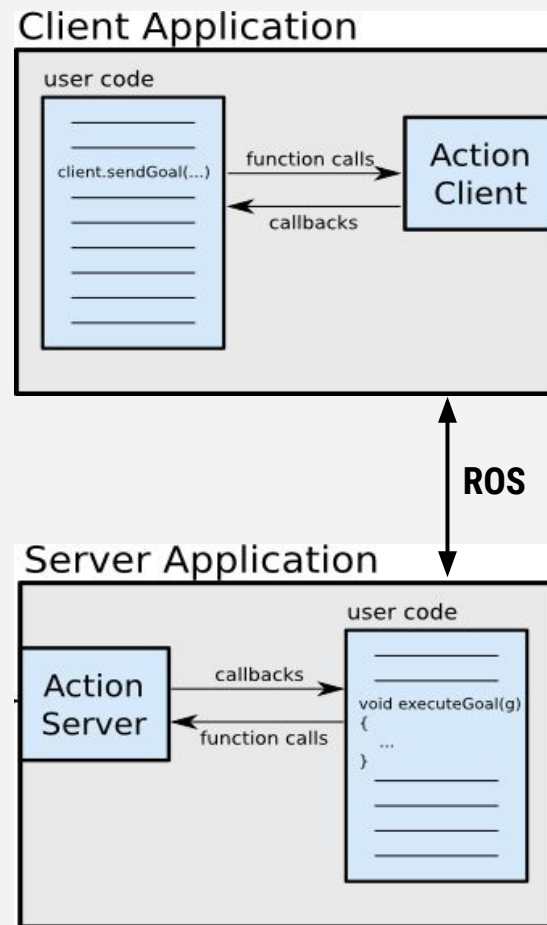
ДЕЙСТВИЯ (ACTIONS)



ДЕЙСТВИЯ (ACTIONS)

<http://wiki.ros.org/actionlib>

- ❑ Пакет actionlib предоставляет API для создания серверной и клиентской части действий (actions)
- ❑ Определение действия состоит из трех частей:
 - ❑ Цели (Goal)
 - ❑ Обратной связи (Feedback)
 - ❑ Результата (Result)
- ❑ Все три поля могут иметь одно или несколько полей следующих типов:
 - ❑ встроенный тип (например, `float64`)
 - ❑ тип уже существующего сообщения (`geometry_msgs/Quaternion`)
 - ❑ массив фиксированной или динамической длины (`float64[]` или `float64[9]`)



ФАЙЛ .ACTION

<http://wiki.ros.org/actionlib>

- ❑ Хранится в директории `/action` пакета
- ❑ Требует зависимости от `actionlib_msgs` в `CMakeLists.txt` и `package.xml` (аналогично зависимости `message_generation` при генерации сообщений)
- ❑ Используется для генерации следующих файлов, которые используются внутри `actionlib` для коммуникации между сервером и клиентом:
 - ❑ `DoDishesAction.msg`
 - ❑ `DoDishesActionGoal.msg`
 - ❑ `DoDishesActionResult.msg`
 - ❑ `DoDishesActionFeedback.msg`
 - ❑ `DoDishesGoal.msg`
 - ❑ `DoDishesResult.msg`
 - ❑ `DoDishesFeedback.msg`

```
./action/DoDishes.action
```

```
# Define the goal
```

```
uint32 dishwasher_id
```

```
# Specify which dishwasher we want  
to use
```

```
---
```

```
# Define the result
```

```
uint32 total_dishes_cleaned
```

```
---
```

```
# Define a feedback message
```

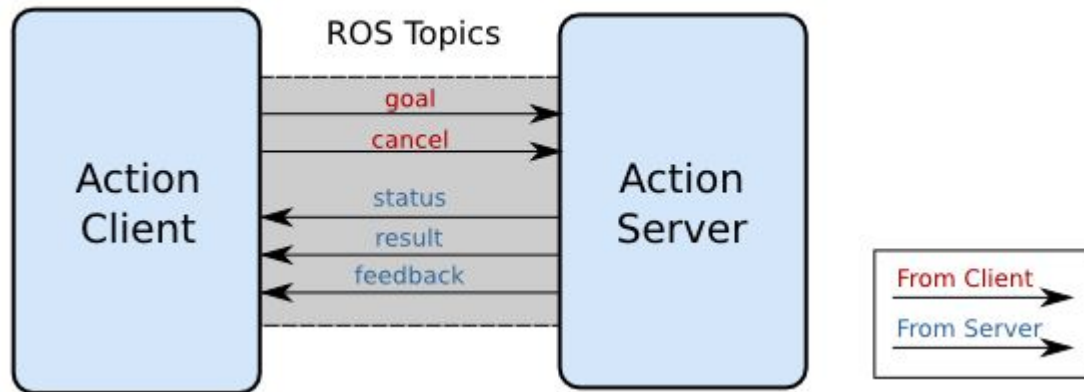
```
float32 percent_complete
```

ДЕЙСТВИЯ

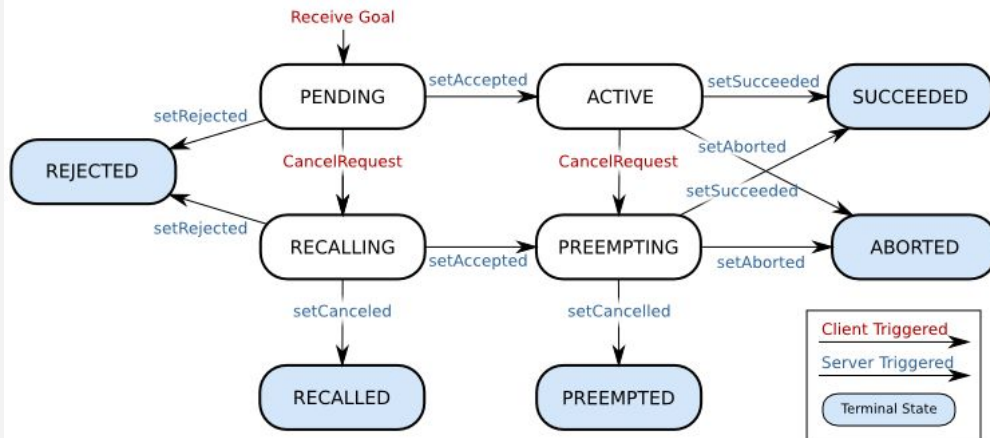
<http://wiki.ros.org/actionlib/DetailedDescription>

- ❑ `goal` — используется для передачи задач в сервер
- ❑ `cancel` — запрос на отмену задачи в сервер
- ❑ `status` — передачи состояния задачи от сервера клиенту (см. список состояний)
- ❑ `feedback` — передача обратной связи о выполнении задачи от сервера к клиенту
- ❑ `result` — используется для однократной отправки результата обработки запроса от сервера клиенту

Action Interface



Server State Transitions



СЕРВЕР ПАРАМЕТРОВ

<http://wiki.ros.org/Parameter%20Server>

Parameter Server — словарь параметров, доступный всем нодам в системе. Используется для хранения различных параметров и доступа к ним в режиме реального времени. *Запускается внутри **rosmaster**.*

Типы данных, поддерживаемые **Parameter Server** :

- ❑ 32-bit integers
- ❑ booleans
- ❑ strings
- ❑ doubles
- ❑ iso8601 dates
- ❑ lists
- ❑ base64-encoded binary data

Доступ к параметрам осуществляется через клиентские библиотеки (**roscpp**, **rospy**, ...), а также инструмент командной строки **rosparam**.

СЕРВЕР ПАРАМЕТРОВ

<http://wiki.ros.org/Parameter%20Server>

- ❑ Чтение параметров
через rospy API
- ❑ Запись параметров
- ❑ Проверка существования
параметра и удаление параметра

```
global_name = rospy.get_param("/global_name")
relative_name = rospy.get_param("relative_name")
private_param = rospy.get_param('~private_name')
default_param = rospy.get_param('default_param',
                                'default_value')
```

```
# fetch a group (dictionary) of parameters
gains = rospy.get_param('gains')
p, i, d = gains['p'], gains['i'], gains['d']
```

```
# Using rospy and raw python objects
rospy.set_param('a_string', 'baz')
rospy.set_param('~private_int', 2)
rospy.set_param('list_of_floats', [1., 2., 3., 4.])
rospy.set_param('bool_True', True)
rospy.set_param('gains', {'p': 1, 'i': 2, 'd': 3})
```

```
if rospy.has_param('to_delete'):
    rospy.delete_param('to_delete')
```

roslaunch

<http://wiki.ros.org/roslaunch>

<http://wiki.ros.org/roslaunch/Tutorials/Roslaunch%20tips%20for%20larger%20projects>

- ❑ **Проблема:** при разработке больших систем возникает необходимость запуска большого количества нодов, установки их параметров, указания вычислителя на котором они должны запускаться (в случае распределенной системы).
- ❑ **Решение:** roslaunch — инструмент, позволяющий описывать процесс запуска системы в xml - файлах (с расширением `.launch`) и запускать всю систему одной командой.
 - ❑ Автоматически запускает `roscore`
 - ❑ Команда `roslaunch_add_file_check(launch)` в `CMakeLists.txt` позволяет проверить `.launch` файл на наличие типовых ошибок (отсутствие обязательного аргумента и т.п.)

roslaunch

```
<launch>
  <!-- local machine already has a definition by default. This tag overrides the default definition with
    specific ROS_ROOT and ROS_PACKAGE_PATH values -->
  <machine name="local_alt" address="localhost" default="true" ros-root="/u/user/ros/ros/"
ros-package-path="/u/user/ros/ros-pkg" />
  <!-- a basic listener node -->
  <node name="listener-1" pkg="rospy_tutorials" type="listener" />
  <!-- pass args to the listener node -->
  <node name="listener-2" pkg="rospy_tutorials" type="listener" args="-foo arg2" />
  <!-- a respawn-able listener node -->
  <node name="listener-3" pkg="rospy_tutorials" type="listener" respawn="true" />
  <!-- start listener node in the 'wg1' namespace -->
  <node ns="wg1" name="listener-wg1" pkg="rospy_tutorials" type="listener" respawn="true" />
  <!-- start a group of nodes in the 'wg2' namespace -->
  <group ns="wg2">
    <!-- remap applies to all future statements in this scope. -->
    <remap from="chatter" to="hello"/>
    <node pkg="rospy_tutorials" type="listener" name="listener" args="--test" respawn="true" />
    <node pkg="rospy_tutorials" type="talker" name="talker">
      <!-- set a private parameter for the node -->
      <param name="talker_1_param" value="a value" />
      <!-- nodes can have their own remap args -->
      <remap from="chatter" to="hello-1"/>
      <!-- you can set environment variables for a node -->
      <env name="ENV_EXAMPLE" value="some value" />
    </node>
  </group>
</launch>
```

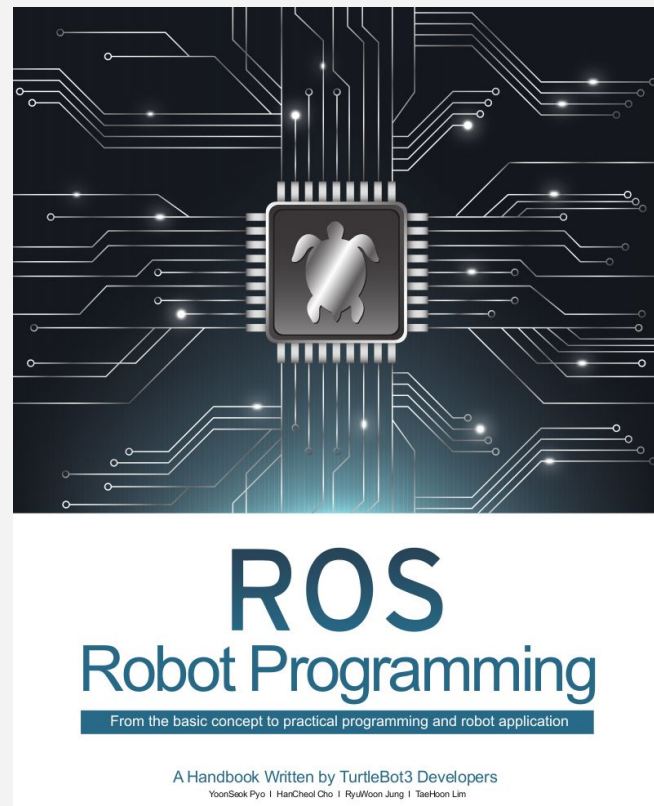
roslaunch. Тэги

<http://wiki.ros.org/roslaunch>

- ❑ `<launch>` – корневой тэг. Обязателен в любом `.launch` файле
- ❑ `<node>` – тэг для запуска нодов.
- ❑ `<machine>` – используется для объявления машин, на которых запускаются ноды. Не используется, если запуск производится локально.
- ❑ `<include>` – используется для подключения других `.xml` файлов в текущий файл
- ❑ `<remap>` – ремапинг аргументов.
- ❑ `<env>` – позволяет установить значение переменной окружения.
- ❑ `<param>` – тэг для установки параметров в `parameter server`.
- ❑ `<rosparam>` – устанавливает параметры в `parameter server` из `.yaml` файла.
- ❑ `<group>` – позволяет применять настройки сразу к группе нодов (например, установить `namespace`).
- ❑ `<test>` – аналогичен `<node>`, но подразумевает запуск нода для тестирования кода других нодов.
- ❑ `<arg>` – установка аргументов запуска.

ДОПОЛНИТЕЛЬНЫЕ ИСТОЧНИКИ

1. Книга: ROS Robot Programming.
YoonSeok Pyo, HanCheol Cho, RyuWoon Jung, TaeHoon Lim (Eng)
2. Обучающие инструкции ROS (Eng)
3. Введение в ROS от Voltbro (Rus)
4. Clearpath Robotics ROS Tutorial (Eng)



ИНФОРМАЦИЯ О ПРЕЗЕНТАЦИИ

Эта презентация была подготовлена Олегом Шипитько в рамках курса “Введение в мобильную робототехнику и Robot Operating System (ROS)” факультета компьютерных наук Высшей Школы Экономики (ВШЭ). Автор выражает благодарность, авторам, чьи материалы были использованы в презентации. В случае, если вы обнаружили в презентации свои материалы, свяжитесь со мной, для включения в список авторов заимствованных материалов.

This presentation was prepared by Oleg Shipitko as part of the “Introduction to Mobile Robotics and Robot Operating System (ROS)” course at the Faculty of Computer Science of Higher School of Economics (HSE). The author is grateful to the authors whose materials were used in the presentation. If you find your materials in a presentation, contact me to be included in the list of contributing authors.