Nodes and topics.

Каждый запущенный файл представляет собой узел – ноду. Ноды общаются через топики. Перед тем как запустить ноду, нужно выполнить команду roscore.

Нода представляет собой самостоятельно исполняемый файл, который заполняет весь терминал.

Чтобы узнать, какие ноды существовали, нужно ввести команду rosnode list. На выходе получится список всех запущенных нод. Если запускать две ноды с одинаковым именем, то первая нода аварийно завершит свою работу.

Чтобы узнать информацию о ноде, необходимо воспользоваться командой rosnode info /(имя ноды).

Топик представляет собой контейнер для хранения данных. Имеет имя(как к нему обращаться) и тип(какие сообщения он в себе хранит). Есть ноды, публикующие сообщения в Топик. Есть ноды, подписанные на этот топик. Ноды «не задумываются» куда переотправляется передаваемая информация.

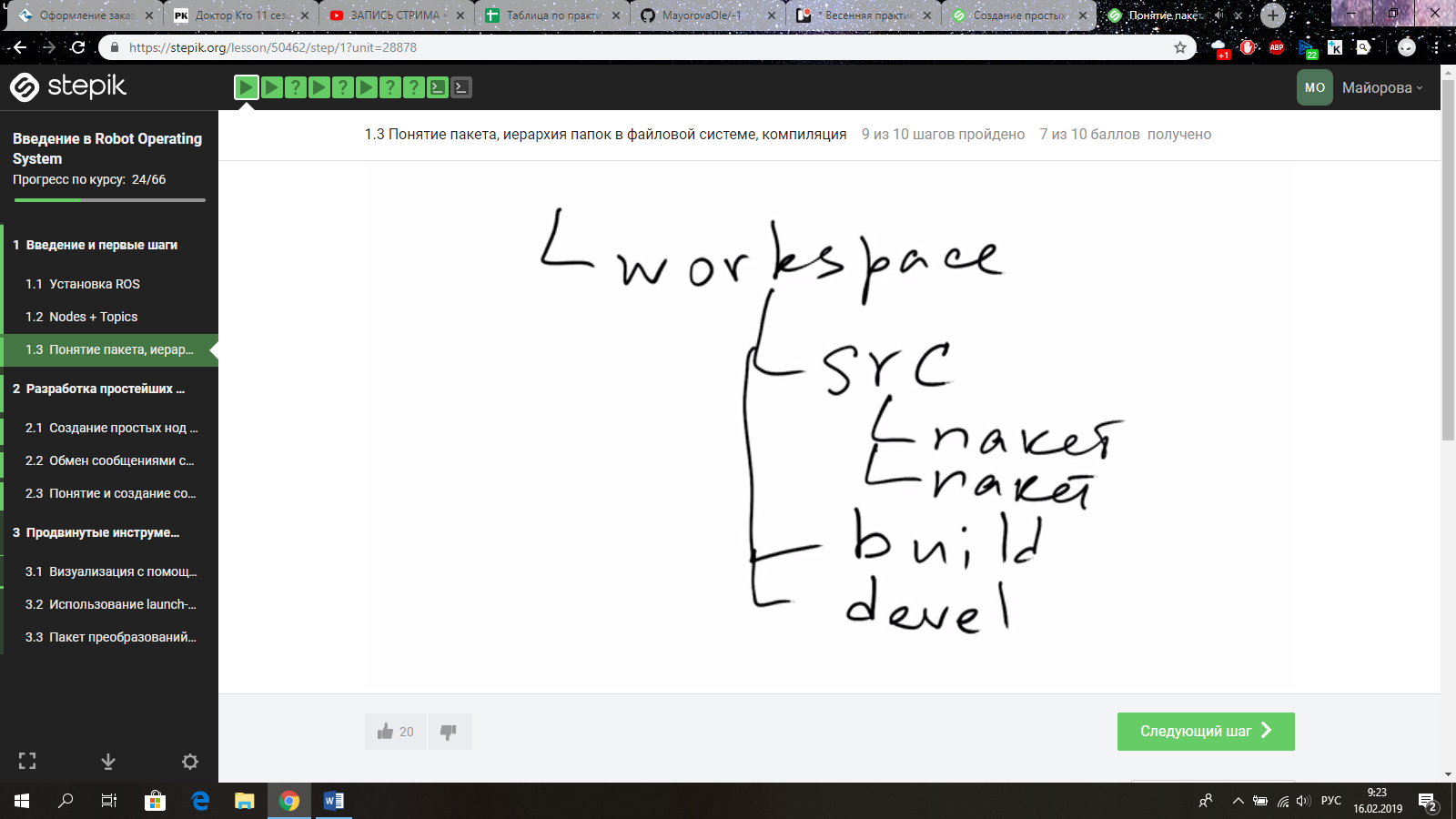
Чтобы получить информацию о топике нужна команда rostopic info /(имя топика). Чтобы узнать больше о типе топика нужна команда rosmsg show (имя типа). Можно сэмулировать передачу сообщения в топик с помощью команды rostopic pub /(имя топика) (тип). Для отслеживания задающих команд используется rostopic echo /(имя топика)

В Рос существует возможность построения графа для передачи взаимоотношений между нодами и топиками. С помощью команды rosrun rqt\_fraph rqt\_graph.

Ещё один способ передачи информации – вызов сервисов. Отличия: ждёт ответа от подписчика, от одного к одному. Сервис состоит из двух частей, одна передаётся подписчику, вторая – обратно. Для того, чтобы увидеть, какие сервисы предоставляются в данный момент системе, используем команду rosservice list. Для того, чтобы больше узнать о сервисе, используем rosservice info/ (имя сервиса). Запуск двух нод порождает запуск одного сервиса для последней из них. Для вызова сервиса используем rosservice call /(имя сервиса).

Понятие пакета, иерархия папок в файловой системе, компиляция.

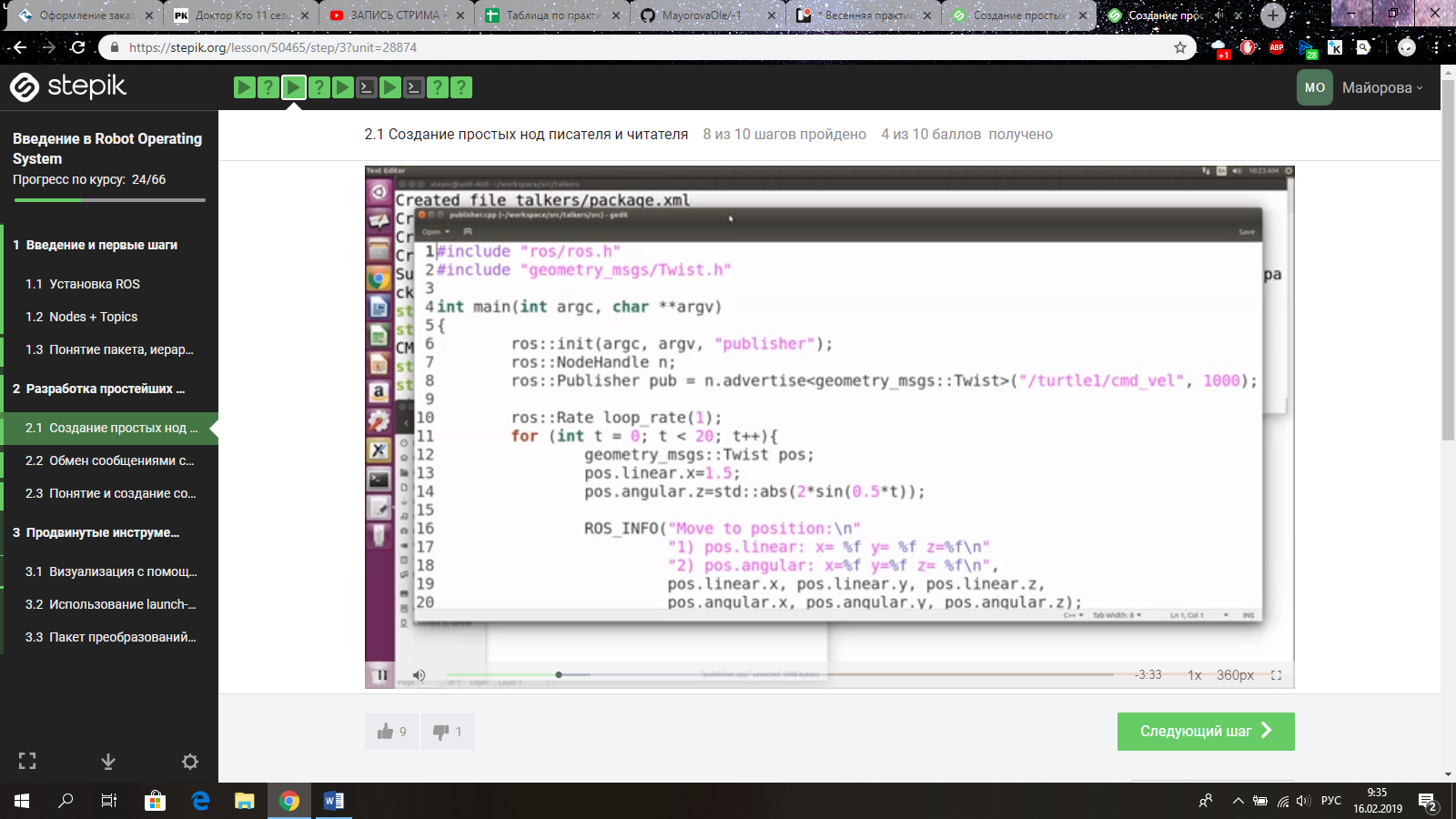
Иерархия папок в росе должна быть строго структурирована.

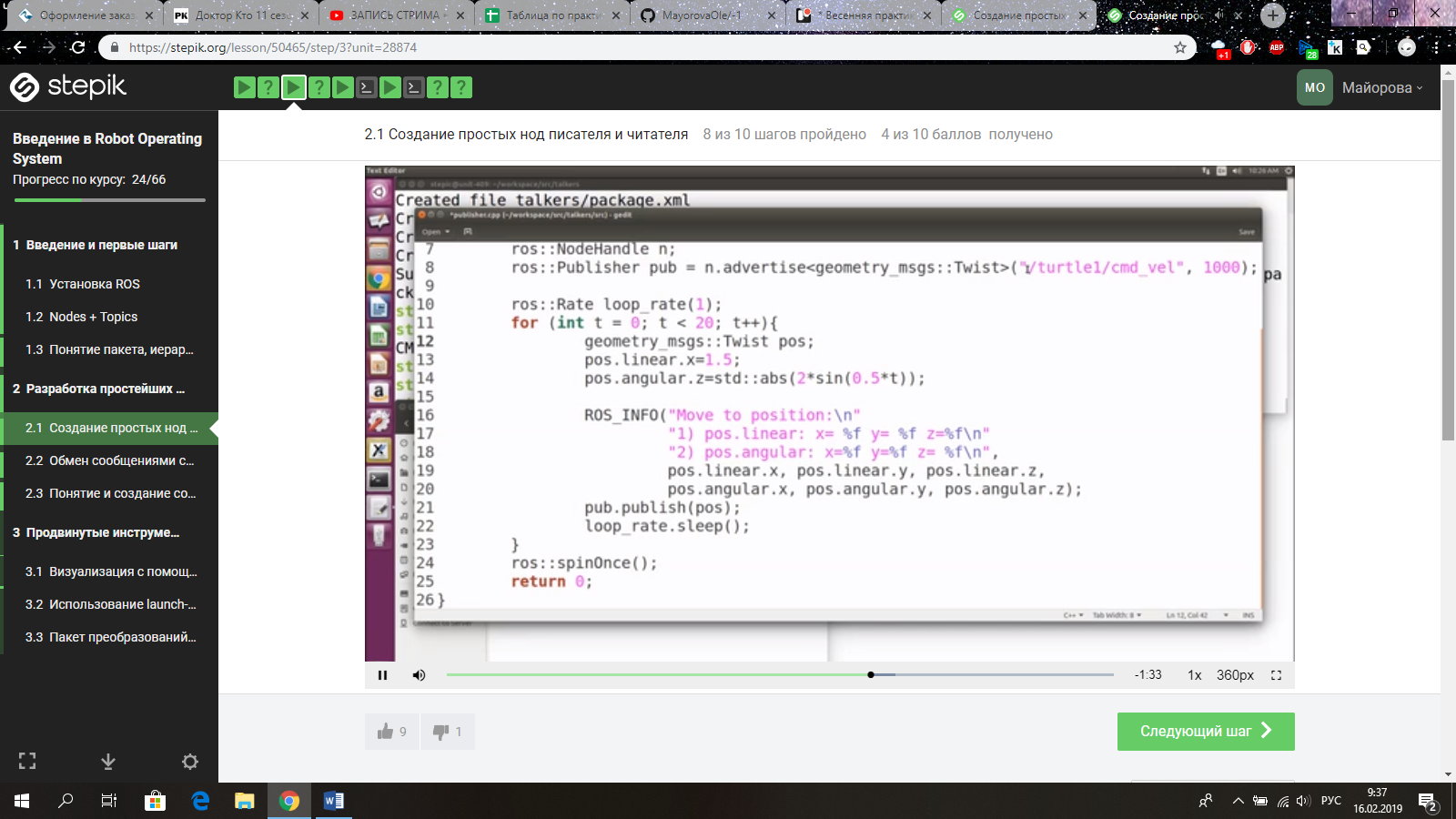


Во время создания пакета после пробела можно указывать зависимости. Для того чтобы собрать пакет, в корневой директории нужно вызвать команду catkin\_make.

***Разработка простейших программ для ros***

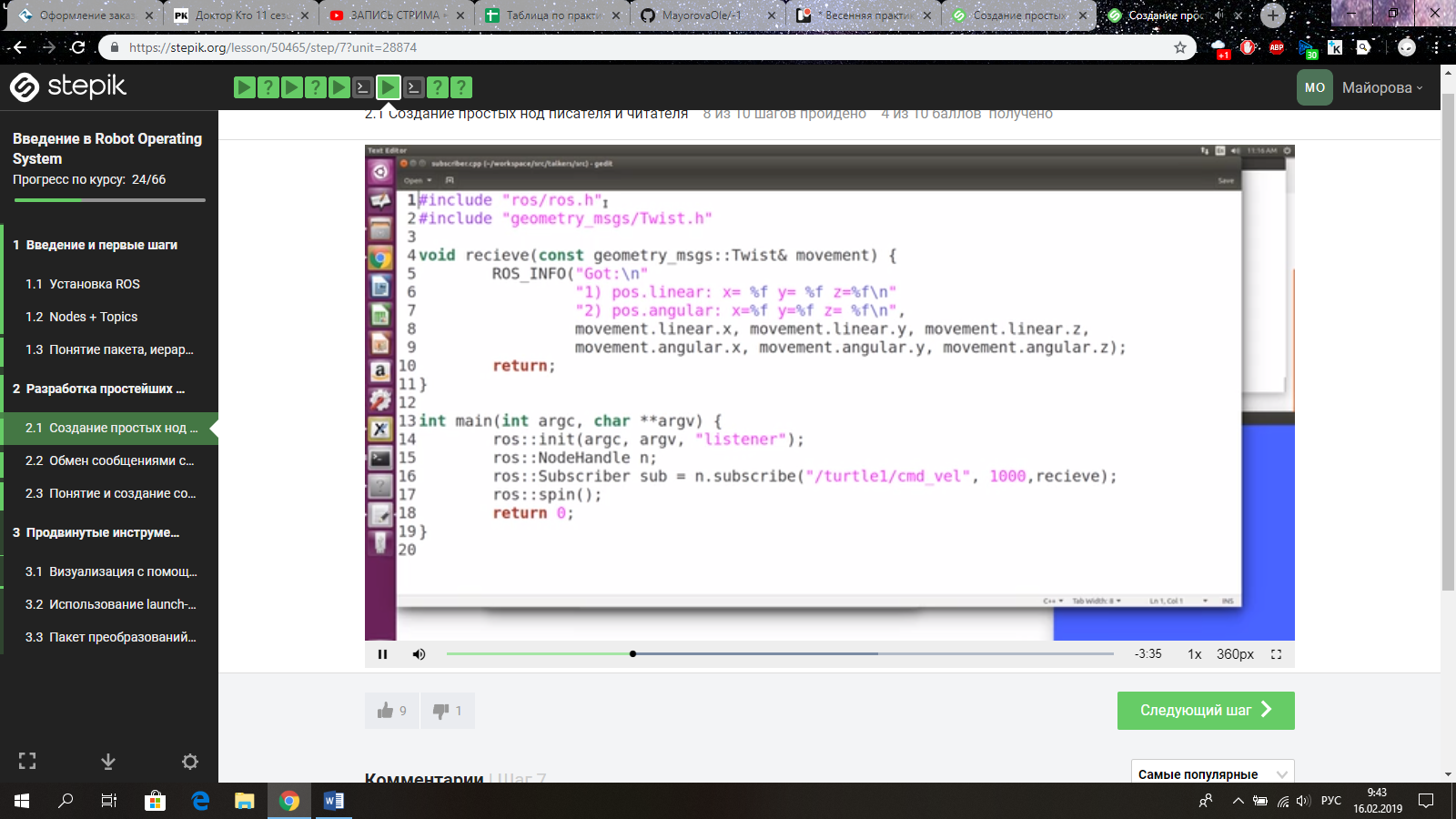
Создание простых нод.

Для того, чтобы писать ноду, нужно создать пакет. Производим необходимую настройку. В папку с пакетом добавляем заранее разработанный код создания ноды, передающей сообщения какому-то топику. 



Чтобы собрать ноду, из корневой папки необходимо запустить catkin\_make. Перед запуском ноды необходимо выполнить команду source devel/setup.bash.

Пример кода для написания ноды, подписанной на какой-то топик.



Обмен сообщениями

Все сообщения, поддерживаемые росом, можно узнать, воспользовавшись командой rosmsg list. Rosmsg show /(имя сообщения) – команда, помогающая получить справку об определённом типе сообщения.

Для реализации обмена сообщениями необходимо воспользоваться командой catkin\_create\_pkg (название пакета). Далее в созданных файлах производим необходимую настройку. При создании файла, отвечающего за собственно передачу сообщений необходимо указать поля данных, которые ей предстоит передавать. Для компиляции необходимо вызвать catkin\_make. В структуре файла, автоматически задаются те типы полей, которые были указаны при создании.

Для вызова программ, передающих сообщения необходимы следующие команды: source devel/setup.bash и после этого можно запускать через rosrun.

Понятие и создание собственного сервис.

Если необходимо компилировать пакеты (например, package1 и package2), которые используют сообщение или сервис, генерируемый в package3, причём скомпилировать package3 отдельно нет возможности, для установки корректного порядка компиляции (предотвращения компиляции пакетов package1 и package2 до пакета package3) в CMakeLists.txt первых пакетов следует после строк

add\_executable(<node\_type> <source\_files> )  
target\_link\_libraries(<node\_type> ${catkin\_LIBRARIES})

следует добавить строку

add\_dependencies(<node\_type> <name\_of\_package3>\_generate\_messages\_cpp)

Эту же строку следует добавить и в CMakeLists.txt пакета package3, если он создаёт ноды, использующие сообщения и сервисы этого пакета.

***Продвинутые инструменты и приёмы.***

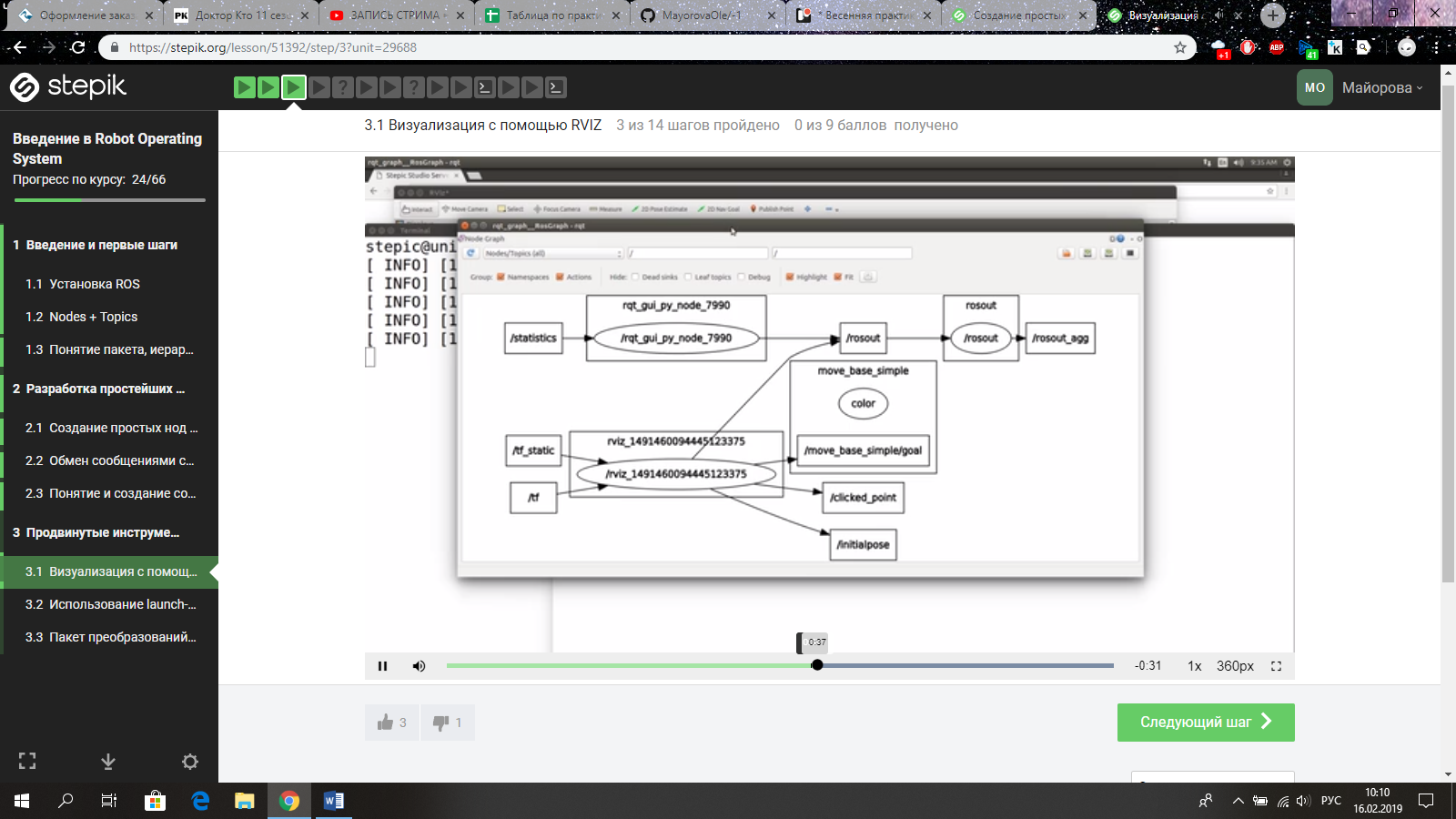
Визуализация с помощью RVIZ

В рос интегрировано средство для визуализации каких-то данных, например, сообщений, передаваемых между нодами.

Для этого существует нода RVIZ. При её запуске появляется графическое окно с полями: слева – поле элементов, которые отображены, справа – настройка вида поля, в центре – само графическое отображение.

В Рос существует возможность построения графа для передачи взаимоотношений между нодами и топиками. С помощью команды rosrun rqt\_fraph rqt\_graph.

Например:



Использование launch-файлов

В ROS существует механизм, позволяющий упростить вызов нескольких нод одновременно. Launch-файл позволяет запускать несколько нод, находящихся в разных пакетах.

Структура такого файла:

Он выглядит как xml.  
Начинается и заканчивается со слов launch.  
внутри можно вызывать различные ноды.

Для вызываемой ноды, есть три обязательных аргумента:

1. Package (pkg= ”имя пакета”)
2. Тип (type=” имя типа”)
3. Имя (name=”Имя ноды”)

Запуск производится из корневой директории, с помощью команды roslaunch.

Пакет преобразования координат TF

В ROS существует пакет, позволяющий выполнять различные математические операции над объектами, заданными в различных системах координат.

Для того чтобы вычислить объект в определённой системе, нам потребуются координаты родительской системы (той самой, в которой хотим получить ответ) и известной.

Возможно широкое применение в решении задач по позиционированию роботов в пространстве.