Петрозаводский государственный университет
Институт математики и информационных технологий
Кафедра информатики и математического обеспечения

Направление подготовки бакалавриата

09.03.04 – Программная инженерия

Профиль направления подготовки бакалавриата

Системное и прикладное программное обеспечение

Отчет о прохождении производственной практики

Преимущества и недостатки ROS и ROS 2

Выполнила:
студентка 2 курса группы 22207

А. Р. Хуснутдинова _____
подпись

Место прохождения практики:
кафедра ИМО, ЦИИ

Период прохождения практики:

_____
период

Руководитель:
О. Ю. Богоявленская

_____
подпись

Итоговая оценка:

_____
оценка

Петрозаводск — 2022

Содержание

1 Введение **3**

2 ROS1 **4**

2.1 Установка ROS1 4

2.2 Создание и сборка ROS пакета 4

3 ROS2 **6**

3.1 Установка ROS2 6

3.2 Создание и сборка ROS пакета 6

4 Сравнение систем **8**

4.1 Преимущества и недостатки ROS1 8

4.2 Преимущества и недостатки ROS2 8

4.3 Сравнительная таблица 9

5 Заключение **10**

1 Введение

ROS1 и ROS2 - опрерационные системы для роботов, наборы библиотек и инструментов, при помощи которых создаются приложения для роботов. А данный момент именно эти системы являются самыми востребованными в своей сфере.

ROS1 была разработана в 2007 году, впоследствии создавались новые, несовместимые друг с другом версии системы. ROS2 - вторая версия ROS1, впервые разработанная в 2017 году.

Целью практики является нахождение достоинств и недостатков обеих систем. Для этого изучим обе системы и изучим некоторые их характеристики.

2 ROS1

2.1 Установка ROS1

Для установки ROS noetic на ubuntu необходима версия 20.04, более поздняя версия не поддерживается. Порядок установки:

1. Настройка компьютера для получения программного обеспечения из packeges.ros.org
2. Настройка ключей (для этого понадобилось установить curl - утилиту для скачивания файлов)
3. Установка пакетов (была выбрана версия $ROS - Base$)
4. Настройка среды для работы с ROS при помощи функции source (необходимо запускать каждый раз при работе из терминала, но можно также добавить в автоматический скрипт)
5. Установка заыисимых пакетов ($python3 - rosdep\ python3 - rosinstall\ python3 - rosinstall - generator\ python3 - wstool\ build - essential$)
6. Установка $python3 - rosdep$ - необходима для упрощение установки системных завиисмостей для ресурсов, которые будут компилироваться а также для некоторых компоненотов ROS

При необходимости можно устанавливать дополнительные пакеты.

2.2 Создание и сборка ROS пакета

Требования к пакету:

1. Пакет должен содержать catkin-совместимый *packege.xml* файл, который содержит метаинформацию о пакете (catkin - система сборки от ROS)
2. Пакет должен содержать $CMakeList.txt$, который использует catkin
3. Каждый пакет должен иметь собственную папку

Создание и сборка пакета:

1. Для начала необходимо создать директорию с названием рабочей среды, которая будет содержать внутри себя директорию *src/*
2. Внутри созданной директории *src/* происходит инициализация рабочей среды с помощью команды $catkin\_init\_workspace$. Данная команда сгенерирует файл $CMakeList.txt$
3. Затем выполняется команда $catkin\_create\_pkg$, на вход которой даётся название для нового пакета, а также прописываются зависимости *rospy* и std_msg. После этого внутри директории *src/* появится директрия с названием пакета, содержащая файлы $CMakrList.txt$ и *packege.xml*, а также директорию *src/*

4. Приступаем к сборке. Для этого в исходной директории запускаем команду $catkin_make$. После сборки пакетов в директории появятся папки *build* и *devel*

5. Пакеты собраны, теперь с помощью функции *rosrun* можно осуществлять необходимые действия над пакетами

3 ROS2

У ROS2 появилась возможность установки на Windows (поддерживается только 10 версия). В качестве операционной системы была выбрана Windows 10.

3.1 Установка ROS2

В качестве дистрибутива для установки был выбран «Humble Hawksbill».

- Для установки понадобились следующие пакеты:
  - Chocolatey - менеджер пакетов для Windows
  - Python 3.8.3
  - Visual Studio
  - Open SSL
  - Qt5
  - Graphviz
- Устнановка CMake и других зависимостей Chocolatey
- Установка RTI Connext DDS для взаимодействия в режиме реального времени
- Настройка среды при помощи команды *call*
- Рабочая среда настроена, взаимодействовать с ней можно при помощи команды *run*

3.2 Создание и сборка ROS пакета

Требования к пакету:

1. Пакет должен содержать файл *packege.xml*, содержащий метаиформацию о пакете
2. Пакет должен содержать файл *setup.py*, содержащий инструкцию по установке пакета
3. Если в пакете содержатся исполняемые файлы, он должен содержать файл *setup.cgf*
4. Пакет должен содержать каталог с именем пакета, каталог используется для поиска пакета инструментами ROS2. Каталог содержит файл $\_\_init.py\_\_$

Шаги создания и сборки пакета:

1. Для начала необходимо создать папку, в которой будет создаваться пакет.
2. Далее с помощью команды *pkg create* и аргумента $--node-name$ создаем пакет с исполняемым файлом типа Hello World.

3. После запуска команды ваш терминал вернет сообщение с информацией о создании пакета, сгенерированных файлов и т.д. (см. Рис.5).

4. Для того, чтобы собрать пакет, необходимо воспользоваться командой $colcon\ build - -merge - install$.

5. Чтобы запустить исполняемый файл, который вы создали с помощью $- - node - name$ аргумента при создании пакета, введите команду: $ros2\ run\ my\_package\ my\_node$. После терминал должен вывести сообщение:

4 Сравнение систем

4.1 Преимущества и недостатки ROS1

Преимущества:

- Финальная версия ROS1 направлена на предоставление Python3 для разработчиков, которым ещё предстоит работать с ROS1
- Несмотря на то, что Python2 более не поддерживается, ROS1, вплоть до финальной версии поддеривает разработку на Python2
- ROS1 поддерживает разработку на Cpp 11/14, юудучи изначально нацеленной на Cpp 98
- Добавлена функциональность *Nodelets*, позволяющая записывать несколько узлов в один и тот же исполняемый файл с обменом данными внутри процесса

Недостатки:

- Система создана 2007 году, добавление новых модификаций в ROS1 требует многожество изменений, которые сделают систему нестабильной
- В 2025 году планируется последняя версия ROS1, после которой не планируется выпуск новых версий
- Для Cpp и Python используются разные, не связанные друг с другом библиотеки, что затруняет разработку на обоих языка
- Отсутствие специфичной структуры, которая сказала бы вам, как следует прописывать функционал узла. Можно добавлять функции обратного вызова в любом месте программы и использовать ООП, но каждая реализация должна быть уникальной
- Использование XML для написания файлов запуска
- Необходимость запустить ROS1 master для запуска узла

4.2 Преимущества и недостатки ROS2

Преимущества:

- ROS 2 позволяет устанавливать безопасные соединения между компонентами системы (нодами);
- взаимодействие осуществляется в режиме «real-time» через концепцию DDS (Data Distributed Services);
- соединение нескольких роботов в одну сеть упрощено;
- реализуется уровень ROS-архитектуры непосредственно на аппаратном обеспечении;
- используются новейшие пакеты обновлений (ROS 1 обновляется не так часто);

Недостатки:

- Использование XML для написания файлов запуска;
- ROS 2 адаптирован не для всех продуктов DDS;

4.3 Сравнительная таблица

Для наглядности сравним системы ROS1 и ROS2, составив таблицу по характеристикам.

Характеристика	ROS1	ROS2
Поддерживаемые ОС	Linux, OS X	Linux, Windows 10, OS X
Поддерживаемые языки программирования	Cpp до 14, Python3, Python2	Python3, Cpp 11, 14, 17, C
Возможнось добавления новых модификаций	Затруднена, так как модификации требуют блшого количества изменений, длающих систему нестабильной	ROS2 была разработана с нуля, поэтому никаких затруднений нет, модификации предусмотрены
Поддержка дистрибутивов	Финальная версия системы будет выпущена в 2025 году, после чего система не будет обновляться	Новые ROS2 версии реализуются каждый год, обеспечивая долгосрочную поддержку
API	Система поддерживает две независимые библиотеки для Cpp и Python	Существует только одна базовая библиотека с именем rcl на C, содержащая все основные функции ROS2. Возможно использование клиентских библиотек поверх rcl
ООП	Отсутствие специфичной структуры, которая бы говорила, как следует прописывать функционал узла	Для написания узла, необходимо создать класс, который наследуется от объекта Node (экономия времени)
Несколько вершин в одном исполняемом файле	Наличие функциональности *Nodelets*	В ROS2 *Nodelets* была непосредственно включена в ядро ROS2 и теперь называется «компонентами».
Файлы запуска	Используется XML	Используется Python, XML

5 Заключение

Таким образом, были изучены системы ROS1 и ROS2. Опытным путём были выявлены различия в установке систем, а также в взаимодействии с ними. После изучения систем, были выявлены некоторые их преимущества и недостатки, а также проведено общее сравнение двух систем.