Колледж Автономной некоммерческой образовательной организации высшего образования

«Научно-технологический университет «Сириус»

Реферат по предмету “Введение в специальность”  
на тему:

**«ROS»**

Работу подготовил:

ФИО, группа

Студент группы К0709-24/1

Черевко Н.С.

Проверил:  
Преподаватель Яковлева С.В.

[Введение](#_61r7u8fo6poa) 3

[1 Основные особенности ROS](#_srkin2ohvm1j) 4

[1.1 Модульность и распределенная архитектура:](#_m2i2mkb9q32o) 4

[1.2 Межплатформенность:](#_9n5a0qw955mn) 4

[1.3 Библиотеки и инструменты:](#_rqwwg9m1dftb) 4

[1.4 Сообщество и открытый исходный код:](#_f7pga3udd765) 5

[1.5 Поддержка различных языков программирования:](#_7u7dqwak7vo0) 5

[2 Применение ROS](#_36jog8es7xj1) 5

[2.1 Научные исследования:](#_q52h0vjqo2mg) 5

[2.2 Промышленность:](#_t0zhmk7yawr3) 6

[2.3 Образование:](#_dm7a7ws5t5a) 6

[2.4 Медицина:](#_gxz7phvhqybm) 6

[3 Преимущества ROS](#_c7ehm1utnlbq) 6

[3.1 Гибкость:](#_utsebgoeawra) 6

[3.2 Скорость разработки:](#_n506j0f48x1w) 7

[3.3 Поддержка сообщества:](#_qvekcd61xxtz) 7

[3.4 Интеграция с другими технологиями:](#_syjaj0kq5e3v) 7

[4.Недостатки ROS](#_quqscmgsh0s5) 7

[4.1 Сложность для новичков:](#_ryxruf3vah6n) 7

[4.2 Ограниченная поддержка реального времени:](#_mtxqm3f1wm8x) 7

[4.3 Зависимость от аппаратного обеспечения:](#_hqy6jhvcxz01) 7

[4.4 Отсутствие централизованной поддержки:](#_1s0r3dugezj9) 7

[5 Заключение](#_jfsba5ys6kx5) 8

[Список использованных источников](#_874eab1bw7fa) 8

# **Введение**

Операционная система для роботов (Robot Operating System, ROS) — это мощная и гибкая платформа для разработки программного обеспечения, предназначенная для создания сложных робототехнических систем. ROS предоставляет разработчикам инструменты, библиотеки и стандарты, которые упрощают процесс создания роботов, позволяя сосредоточиться на реализации функциональности, а не на решении низкоуровневых задач.

ROS была разработана в 2007 году в Стэнфордском университете, а затем продолжена в Willow Garage. Сегодня ROS активно используется в научных исследованиях, промышленности и образовании, став одной из самых популярных платформ для робототехники. Её открытый исходный код, активное сообщество и универсальность делают её незаменимым инструментом для разработки роботов различного назначения.

# **1 Основные особенности ROS**

## *1.1 Модульность и распределенная архитектура*

ROS построена на модульной архитектуре, где каждый компонент (узел) выполняет определённую задачу. Узлы могут обмениваться данными через механизм сообщений, что позволяет создавать сложные системы из простых компонентов. Например, один узел может отвечать за обработку данных с датчиков, а другой — за управление двигателями.

Топики (Topics): Узлы обмениваются данными через топики, которые являются каналами передачи сообщений.

Сервисы (Services): Узлы могут вызывать друг у друга функции через сервисы.

Действия (Actions): Для выполнения длительных задач используются действия, которые позволяют отслеживать прогресс и прерывать выполнение.

## *1.2 Межплатформенность*

ROS поддерживает различные операционные системы, включая Linux, Windows и macOS. Это делает её универсальной платформой для разработки роботов на разных аппаратных платформах.

## *1.3 Библиотеки и инструменты*

ROS предоставляет множество готовых библиотек для работы с датчиками, управления движением, навигации, машинного зрения и других задач. Например:

OpenCV: Для обработки изображений.

MoveIt: Для планирования движений роботов-манипуляторов.

Navigation Stack: Для навигации мобильных роботов.

Также в ROS есть инструменты для визуализации данных (например, Rviz) и симуляции (Gazebo), которые упрощают тестирование и отладку.

## *1.4 Сообщество и открытый исходный код*

ROS является открытой платформой, что способствует активному развитию сообщества. Пользователи ROS могут делиться своими наработками через репозитории, такие как GitHub. Это ускоряет процесс разработки и внедрения новых технологий.

## *1.5 Поддержка различных языков программирования*

ROS поддерживает несколько языков программирования, включая C++ и Python. Это делает её доступной для разработчиков с разным уровнем опыта.

# 

# **2 Применение ROS**

## *2.1 Научные исследования*

ROS широко используется в университетах и исследовательских центрах для разработки прототипов роботов. Например, ROS применяется в проектах, связанных с автономными транспортными средствами, дронами, роботами-манипуляторами и гуманоидными роботами.

Autonomous Vehicles: ROS используется для навигации, планирования маршрутов и обработки данных с датчиков.

Humanoid Robots: Например, робот Atlas от Boston Dynamics использует ROS для управления.

## *2.2 Промышленность*

сортировка, транспортировка и складирование. Компании внедряют ROS для создания гибких и адаптивных робототехнических систем, которые могут работать в изменяющихся условиях.

KUKA: ROS испольВ промышленности ROS используется для автоматизации процессов, таких как сборка, зуется для управления промышленными роботами-манипуляторами.

Universal Robots: ROS интегрирован в cobots (коллаборативные роботы).

## *2.3 Образование*

ROS активно используется в образовательных программах для обучения студентов основам робототехники. Благодаря своей модульности и открытости, ROS позволяет студентам быстро осваивать ключевые концепции и применять их на практике.

## *2.4 Медицина*

## ROS также применяется в медицинской робототехнике, например, для разработки роботов-ассистентов в хирургии или реабилитационных роботов.

# 

# **3 Преимущества ROS**

## *3.1 Гибкость*

ROS позволяет разрабатывать роботов для различных задач, от простых мобильных платформ до сложных промышленных роботов.

## *3.2 Скорость разработки*

Благодаря готовым библиотекам и инструментам, разработчики могут быстро создавать прототипы и тестировать новые идеи.

## *3.3 Поддержка сообщества*

Активное сообщество ROS обеспечивает постоянное обновление платформы и предоставляет поддержку новым пользователям.

## *3.4 Интеграция с другими технологиями*

ROS легко интегрируется с другими технологиями, такими как машинное обучение, компьютерное зрение и IoT.

# 

# **4.Недостатки ROS**

## *4.1 Сложность для новичков*

ROS имеет высокий порог входа, особенно для тех, кто не знаком с Linux и программированием на C++ или Python.

## *4.2 Ограниченная поддержка реального времени*

ROS не предназначена для систем, требующих жесткого реального времени, что может быть проблемой для некоторых промышленных приложений.

## *4.3 Зависимость от аппаратного обеспечения*

Некоторые функции ROS могут требовать специфического оборудования, что увеличивает стоимость разработки.

## *4.4 Отсутствие централизованной поддержки*

Хотя сообщество ROS активно, отсутствие централизованной поддержки может затруднить решение сложных проблем.

Примеры использования ROS:

PR2 (Willow Garage):

PR2 — это робот-платформа, разработанная Willow Garage, которая демонстрирует возможности ROS. PR2 может выполнять такие задачи, как уборка, доставка предметов и взаимодействие с людьми.

Autonomous Vehicles:

ROS используется в проектах автономных транспортных средств, таких как автомобили и дроны. Например, ROS применяется в навигации, планировании маршрутов и обработке данных с датчиков.

Industrial Robots:

В промышленности ROS используется для управления роботами-манипуляторами, такими как KUKA и Universal Robots.

# 

# **5 Заключение**

ROS является мощной и гибкой платформой для разработки робототехнических систем. Её модульность, открытый исходный код и поддержка сообщества делают её идеальным инструментом для научных исследований, промышленности и образования. Несмотря на некоторые ограничения, ROS продолжает развиваться, предлагая новые возможности для создания инновационных роботов.

Будущее ROS связано с дальнейшей интеграцией с технологиями искусственного интеллекта, машинного обучения и IoT, что откроет новые горизонты для робототехники.

# 

# **Список использованных источников**

1. Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., ... & Ng, A. Y. (2009). ROS: an open-source Robot Operating System. ICRA workshop on open source software, 3(3.2), 5.
2. ROS.org. (n.d.). Official ROS documentation. Retrieved from https://www.ros.org/
3. Gazebo. (n.d.). Official Gazebo documentation. Retrieved from http://gazebosim.org/
4. Siciliano, B., & Khatib, O. (2016). Springer Handbook of Robotics. Springer.
5. Koubaa, A. (2017). Robot Operating System (ROS): The Complete Reference. Springer.
6. Willow Garage. (n.d.). PR2 robot. Retrieved from https://www.willowgarage.com/
7. Open Robotics. (n.d.). ROS Industrial. Retrieved from https://rosindustrial.org/
8. Boston Dynamics. (n.d.). Atlas robot. Retrieved from https://www.bostondynamics.com/atlas
9. KUKA. (n.d.). Industrial robots. Retrieved from https://www.kuka.com/
10. Universal Robots. (n.d.). Collaborative robots. Retrieved from https://www.universal-robots.com/