**Лабораторная работа №2**

**Выполнила:** Исламова Сания

**Группа:** НПИмд-01-24

**Задачи:**

* Установить ROS2 Humble
* Изучить планировщик Pyperplan <https://github.com/aibasel/pyperplan>
* Построить модель среды с tb3 (4) с манипулятором, либо любой другой колесный робот с манипулятором <https://emanual.robotis.com/docs/en/platform/turtlebot3/machine_learning/>
* Создать ROS узел с планировщиком.
* Запуск (установка) VirtualBox
* Установка Linux Ubuntu 22.04 на виртуальную машину

Проверка версии операционной системы

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

**Установка ROS2**

Ссылка на документацию: <https://docs.ros.org/en/humble/Installation/Ubuntu-Install-Debs.html>

Проверка и настройка локали системы на UTF-8:

*locale # проверка UTF-8*

*sudo apt update && sudo apt install locales*

*sudo locale-gen en\_US en\_US.UTF-8*

*sudo update-locale LC\_ALL=en\_US.UTF-8 LANG=en\_US.UTF-8*

*export LANG=en\_US.UTF-8*

*locale # проверка настроек*

Эти две команды используются для управления репозиториями программного обеспечения в системе Linux:

*sudo apt install software-properties-common* (Этот пакет предоставляет инструменты для управления репозиториями программного обеспечения, в том числе для их добавления и удаления.)

*sudo add-apt-repository universe* (Эта команда добавляет репозиторий «universe» в список источников программного обеспечения вашей системы.)

Добавление GPG-ключа:

*sudo apt update && sudo apt install curl -y* ( обновляет локальный индекс пакетов вашего менеджера пакетов APT, Устанавливает инструмент командной строки curl. curl используется для передачи данных с сервера или на сервер по различным протоколам (например, HTTP). Флаг -y автоматически отвечает «да» на любые запросы во время установки)

*sudo curl -sSL* [*https://raw.githubusercontent.com/ros/rosdistro/master/ros.key -o /usr/share/keyrings/ros-archive-keyring.gpg*](https://raw.githubusercontent.com/ros/rosdistro/master/ros.key%20-o%20/usr/share/keyrings/ros-archive-keyring.gpg)(Эта команда загружает ключ GPG и сохраняет его в связке ключей вашей системы, этот ключ используется для проверки подлинности пакетов ROS, чтобы предотвратить установку вредоносного программного обеспечения).

*$("deb [arch=echodpkg --print-architecture) signed-by=/usr/share/keyrings/ros-archive-keyring.gpg] http://packages.ros.org/ros2/ubuntu $(. /etc/os-release && echo $UBUNTU\_CODENAME) main" | sudo tee /etc/apt/sources.list.d/ros2.list > /dev/null*

(Эта команда добавляет новый репозиторий для пакетов ROS 2 в список источников APT - Advanced Package Tool — это бесплатная система управления пакетами программного обеспечения)

Обновляем кэши репозитория apt после настройки репозиториев, а также установкой новых пакетов всегда рекомендуется обновить систему.

*sudo apt update*

*sudo apt upgrade*

Установка ROS2 Humble:

*sudo apt install ros-humble-desktop-full* (Эта команда устанавливает «полную настольную» версию ROS)

*sudo apt install ros-humble-ros-base* (эта команда устанавливает основные компоненты ROS Humble. Она установит необходимые инструменты и библиотеки для начала работы с ROS)

*sudo apt install ros-dev-tools* (эта команда для установки инструментов разработки)

*ros2 run demo\_nodes\_cpp talker* (запускает простой узел ROS 2, который публикует сообщения)

*ros2 run demo\_nodes\_py listener* (Если узел talker также запущен и публикует сообщения, узел listener будет получать эти сообщения и выводить их на консоль)

Изображение выглядит как текст, снимок экрана, Мультимедийное программное обеспечение, программное обеспечение

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Мультимедийное программное обеспечение, программное обеспечение

Автоматически созданное описание

**Установка Pyperplan**

Ссылка на документацию: <https://pypi.org/project/pyperplan/>

*pip install pyperplan*

Исполняемый файл pyperplan принимает два аргумента: файл домена PDDL и файл задачи PDDL.

*pyperplan benchmarks/tpp/domain.pddl benchmarks/tpp/task01.pddl* (**Файл домена (.pddl):** этот файл описывает общие характеристики области планирования, **Файл задачи (.pddl):** этот файл описывает конкретный пример задачи планирования в рамках области, определённой в файле области)

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

**Построить модель среды с tb3 (4) с манипулятором, либо любой другой колесный робот с манипулятором**

**Установка Turtlebot3**

Ссылка на документацию по установке turtlebot3: <https://emanual.robotis.com/docs/en/platform/turtlebot3/quick-start/>

*$ sudo apt install ros-humble-gazebo-\** (устанавливает набор пакетов, которые предоставляют необходимые инструменты и библиотеки для интеграции ROS с Gazebo. Это позволяет моделировать роботов и окружающую среду в Gazebo и управлять ими с помощью ROS.)

*$ sudo apt install ros-humble-cartographer* (Эта команда устанавливает основной пакет Cartographer.)

*$ sudo apt install ros-humble-cartographer-ros* (Это устанавливает оболочку или мост ROS для Cartographer. Этот пакет интегрирует библиотеку Cartographer с платформой ROS)

*$ sudo apt install ros-humble-navigation2* (Это устанавливает основной navigation2 метапакет. Метапакет — это набор связанных между собой пакетов. Эта команда устанавливает основные библиотеки, алгоритмы и узлы, из которых состоит стек Navigation2.)

*$ sudo apt install ros-humble-nav2-bringup* (Этот пакет предоставляет файлы запуска и конфигурации для простого запуска и работы базовой навигационной системы. )

*$ source ~/.bashrc* (Эта команда обращается к вашему файлу .bashrc . Это обновляет среду оболочки с учётом любых изменений, внесённых в файл конфигурации)

*$ sudo apt install ros-humble-dynamixel-*sdk (Эта команда устанавливает пакет Dynamixel SDK ROS. Dynamixel — популярная серия сервоприводов, используемых в робототехнике. Этот пакет предоставляет драйверы и инструменты для управления двигателями Dynamixel в среде ROS)

*$ sudo apt install ros-humble-turtlebot3-msgs* (Эта команда устанавливает определения сообщений TurtleBot3. Этот пакет содержит типы сообщений, которые ROS использует для взаимодействия с роботом TurtleBot3. Он необходим для правильного взаимодействия с роботом и получения обратной связи от датчиков.)

*$ sudo apt install ros-humble-turtlebot3* (Эта команда устанавливает пакет TurtleBot3 ROS. Этот пакет содержит драйверы, файлы конфигурации и запуска, которые упрощают взаимодействие с роботом TurtleBot3.)

*$ mkdir -p ~/turtlebot3\_ws/src*

*$ cd ~/turtlebot3\_ws/src/*

*$ git clone -b humble-devel https://github.com/ROBOTIS-GIT/DynamixelSDK.git*

*$ git clone -b humble-devel https://github.com/ROBOTIS-GIT/turtlebot3\_msgs.git*

*$ git clone -b humble-devel https://github.com/ROBOTIS-GIT/turtlebot3.git*

*$ cd ~/turtlebot3\_ws*

*$ colcon build --symlink-install* (Этот параметр создаёт символические ссылки вместо копирования файлов во время установки.)

*$ echo 'source ~/turtlebot3\_ws/install/setup.bash' >> ~/.bashrc*

*$ source ~/.bashrc*

*$ echo 'export ROS\_DOMAIN\_ID=30 #TURTLEBOT3' >> ~/.bashrc* (Переменная среды ROS\_DOMAIN\_ID используется в ROS 2 для указания идентификатора домена для связи. Это особенно важно, когда одновременно работает несколько узлов ROS 2, особенно в ситуациях, когда нескольким роботам или системам необходимо взаимодействовать, но они не должны мешать друг другу.)

*$ source ~/.bashrc*

*cd ~/turtlebot3\_ws/src/*

*git clone -b humble-devel https://github.com/ROBOTIS-GIT/turtlebot3\_simulations.git*

*cd ~/turtlebot3\_ws && colcon build --symlink-install* (Пакет turtlebot3\_simulations содержит смоделированные среды для TurtleBot3. Это позволяет запускать и тестировать код TurtleBot3 в смоделированном мире, что значительно быстрее и проще, чем работа в реальном мире.)

*ros2 launch turtlebot3\_gazebo turtlebot3\_world.launch.py* (Заупуск робота в пустом мире Gazebo)

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

*sudo apt install ros-humble-dynamixel-sdk ros-humble-ros2-control ros-humble-ros2-controllers ros-humble-gripper-controllers ros-humble-moveit* (Эти пакеты необходимы для управления роботом, особенно в контексте TurtleBot3 с роботизированной рукой)

*cd ~/turtlebot3\_ws/src/*

*git clone -b humble-devel* [*https://github.com/ROBOTIS-GIT/turtlebot3\_manipulation.git*](https://github.com/ROBOTIS-GIT/turtlebot3_manipulation.git)

*cd ~/turtlebot3\_ws && colcon build --symlink-install*

*ros2 launch turtlebot3\_manipulation\_bringup gazebo.launch.py* (Запуск среды с манипулятором)

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

**Ссылка на видео «Открытие пустого мира и установка манипуляторной руки в Gazebo»:** <https://disk.yandex.ru/i/M_TdEujrj4rKtQ>

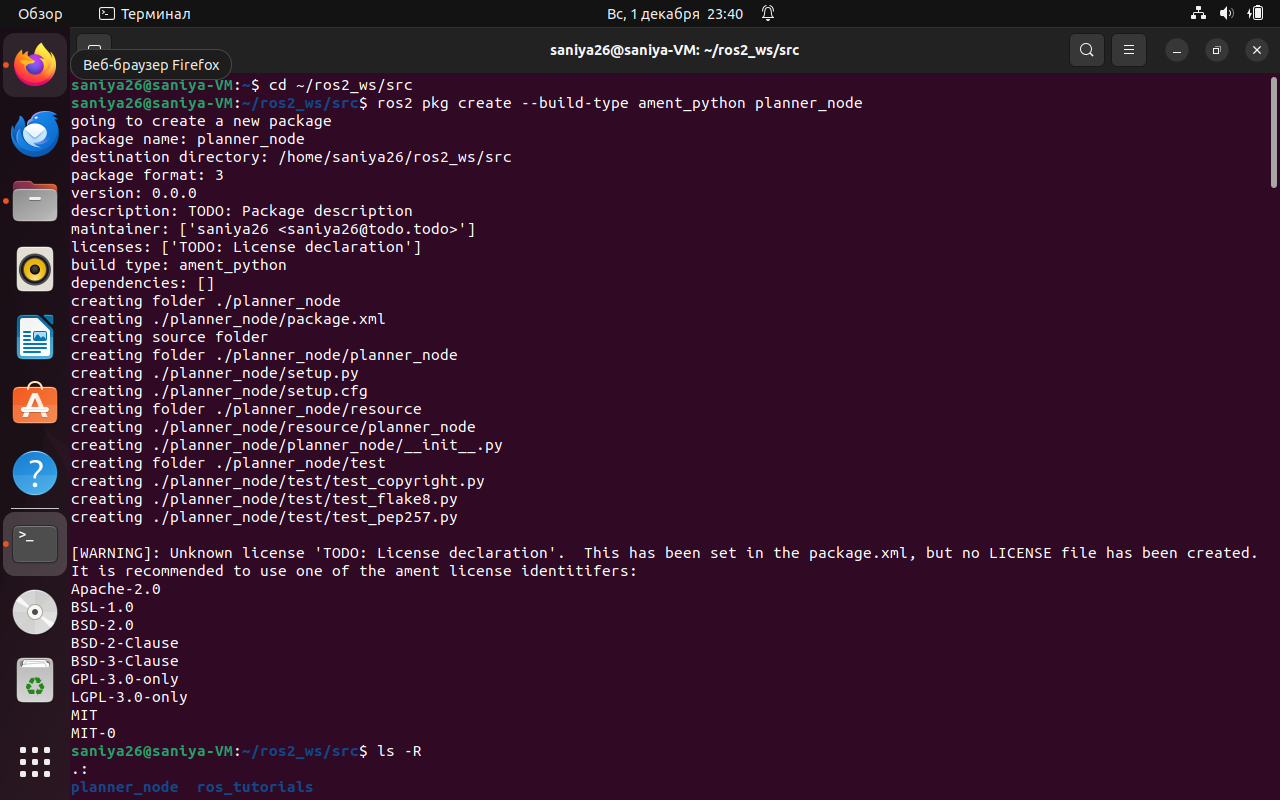
Это позволяет немедленно получить доступ к обновлённому файлу .bashrc и сделать переменные среды рабочей области доступными в текущем терминале.

**Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание**

**Создание ROS-узла**

Создание нового пакета ROS2 и его структура

****

**Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание**

Реализация узла с использованием Pyperplan

Изображение выглядит как текст, электроника, снимок экрана, программное обеспечение

Автоматически созданное описание

Создание файлов domain.pddl и problem.pddl

PDDL - Planning Domain Definition Language. Файлы domain.pddl и problem.pddl используются в классическом планировании с помощью языка определения предметной области планирования (PDDL). Они описывают задачу планирования.

*domain.pddl (Определение домена):* этот файл определяет общие характеристики домена планирования. Это многократно используемое описание структуры среды, действий и объектов, которое можно применять в различных задачах.

*problem.pddl (Пример задачи):* этот файл определяет конкретный пример задачи планирования в рамках области, определённой в domain.pddl

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

Сборка рабочей среды и запуск узла

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Отправка задач. Узел подписывается на тему, чтобы обработать путь к файлу примера задач.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Файл для сохранения решения задачи планирования, выполненной с помощью pddl-планировщика. Содержит последовательность действий.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание