Introduction to ROS2: Basics, Motion, and Vision

Geesara Kulathunga

Innopolis University

Course Structure

- Section 1
  - ROS2 Fundamentals
  - Workout Examples
- Section 2
  - ROS2 Debugging
  - ROS2 Visualization
  - ROS2 Basic Control
  - ROS2 Simulation (Gazebo+SDF)
- Section 3
  - Mini Group Project
- References
  - https://docs.ros.org/en/foxy/index.html

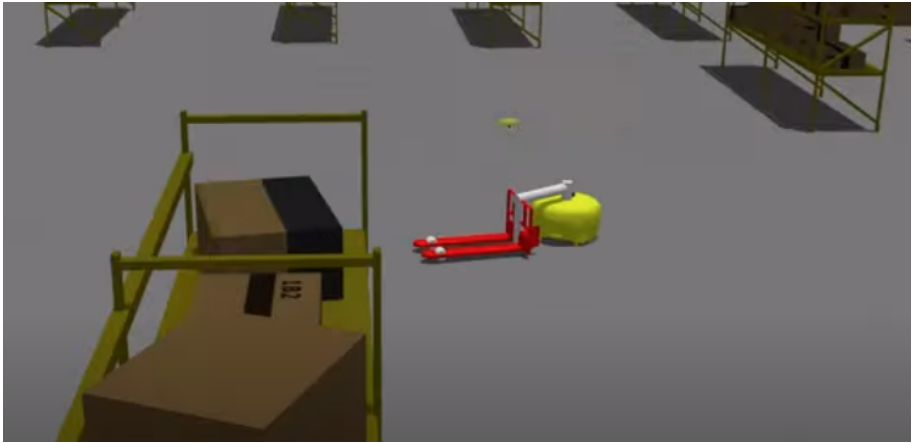
Course Logistics

1. 05.06.2023 (17:50-21:00) section 01
2. 07.06.2023 (17:50-21:00) section 01
3. 12.06.2023 (17:50-21:00) lab 01, hw 01
4. 14.06.2023 (17:50-21:00) section 02
5. 19.06.2023 (17:50-21:00) lab 02, hw 02
6. 20.06.2023 (17:50-19:20) section 02
7. 26.06.2023 (17:50-21:00) lab 03, hw 03, releasing the mini project
8. 28.06.2023 (17:50-21:00) no class, project time
9. 03.07.2023 (17:50-21:00) no class, project time
10. 05.07.2023 (17:50-21:00) project presentation
11. 10.07.2023 (17:50-19:20) project presentation

Course Evaluation

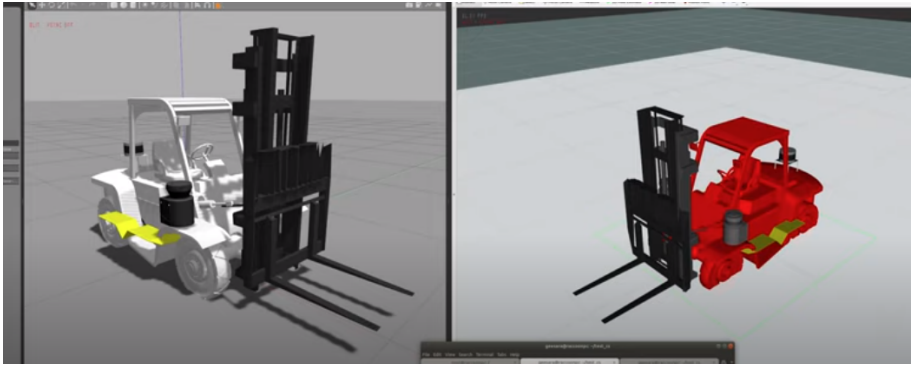
- In-class activities (25% + 5%)
- Mini-project 30%
- Homework 40%

Why do we need ROS?



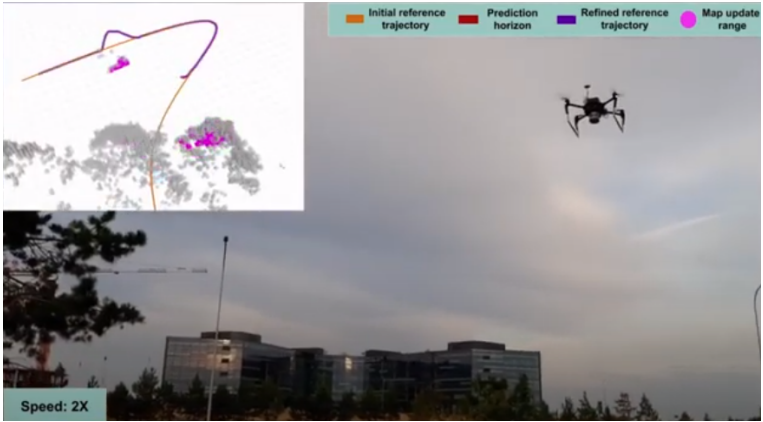
https://youtu.be/wU0WEoQxzTo

Why do we need ROS?



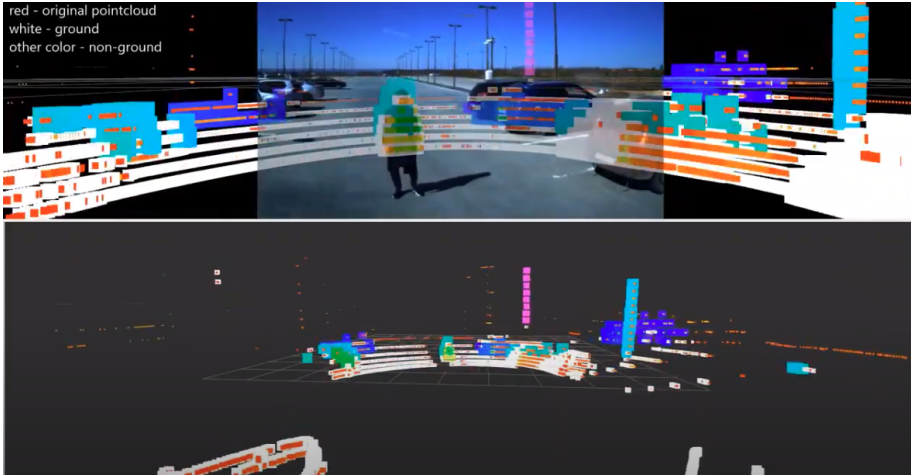
https://youtu.be/Tk5wmhEv96I

Why do we need ROS?



https://youtu.be/_sLVYOvMns0

Why do we need ROS?



https://youtu.be/iHd_ZkhKPjc

In Robotics, before 2007

1. Problem of reusability
2. Which standard to follow
3. Multiple drivers for the same device, write internal communications, e.g., ICP (Inter Process Communication), managing of shared data
4. Implements the same algorithm in different standards
5. Have to start from the scratch most of the time

ROS (Robot Operating System)

1. Begun at the Stanford Artificial Intelligence Laboratory in 2007
2. Today it is being used by many organizations
3. OS (Operating System) manages communications between computer software and hardware. Framework is abstraction layer in which it provides generic functionality. User is able to select which models or components are to enable to achieve the necessity
4. ROS is a meta operating system, i.e., does not provide full capabilities that a operating system provides. However, it build on top a OS, it can be seen as abstraction for accessing underling all the software and hardware layers.

ROS (Robot Operating System)

ROS Fuerte Turtle	April 23, 2012			--
ROS Electric Emys	August 30, 2011			--
ROS Diamondback	March 2, 2011			--
ROS C Turtle	August 2, 2010			--
ROS Box Turtle	March 2, 2010			--

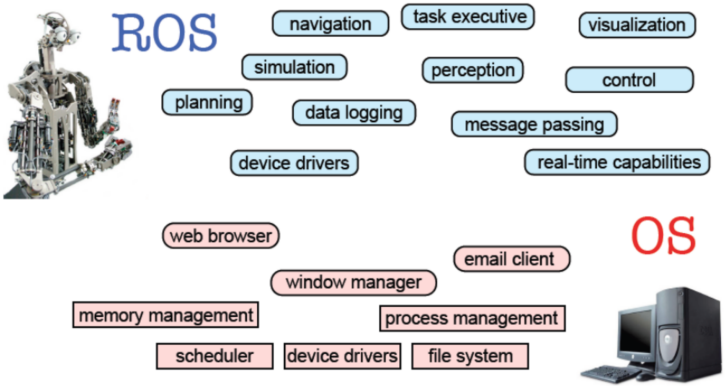
ROS (Robot Operating System)

ROS Noetic Ninjemys **(Recommended)**	May 23rd, 2020			May, 2025 (Focal EOL)
ROS Melodic Morenia	May 23rd, 2018			May, 2023 (Bionic EOL)
ROS Lunar Loggerhead	May 23rd, 2017			May, 2019
ROS Kinetic Kame	May 23rd, 2016			April, 2021 (Xenial EOL)
ROS Jade Turtle	May 23rd, 2015			May, 2017

ROS Philosophy

1. Peer-to-peer (communicate over defined API, e.g., messages, services) network, e.g., master (ROS1), nodes, and parameter server(ROS1). A Computation Graph contains ROS2 nodes that communicate with each other no need of a **master**
2. Has its own file system, e.g., package.xml, messages and service types
3. Distributed (the whole program can split into several sub-programs and run on multiple computers)
4. Multi-lingual (sub-programs can be written in different languages (C++, Python, Java))
5. Light-weight (Standalone libraries are wrapped around with a thin layer)
6. Open-source

ROS vs. OS



@https://web2.qatar.cmu.edu/~gdicaro/16311-Fall17/slides/Lab-1-ROS-Intro.pdf

Robots that use ROS

https://robots.ros.org/

Aerial

Component

Ground

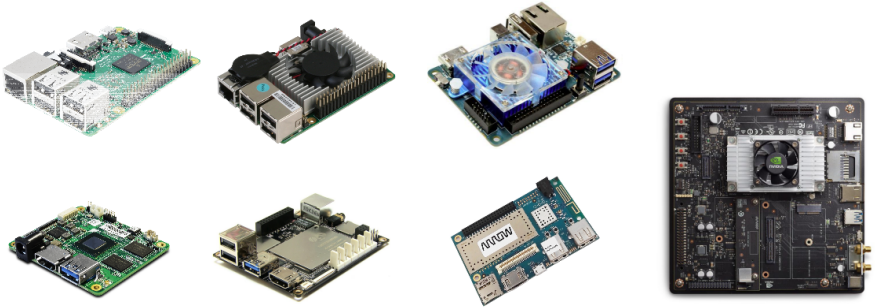
Manipulator

Marine

Sensing Capabilities

http://wiki.ros.org/Sensors

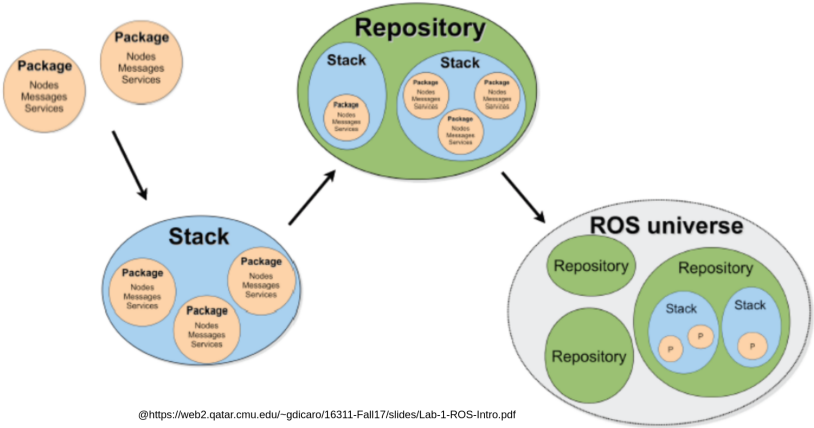
Computing Capabilities



ROS Building Blocks

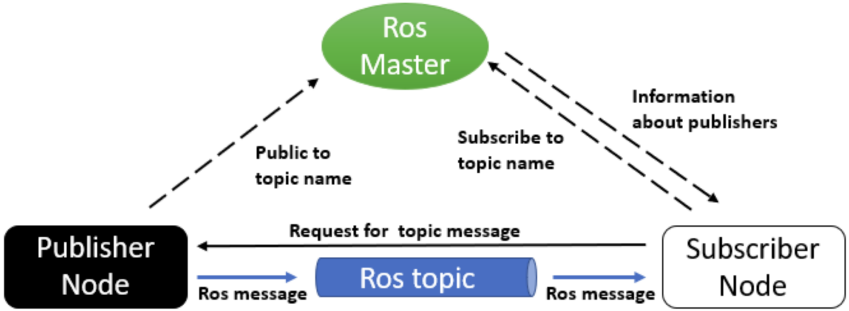
1. **Master** (ROS1)
2. **Parameter server** (ROS1)
3. Services and Topics
4. Messages
5. Actions(ROS2)
6. Nodes and Nodelts (ROS1), Composition(ROS2)
7. Packages and Stacks

ROS Ecosystem



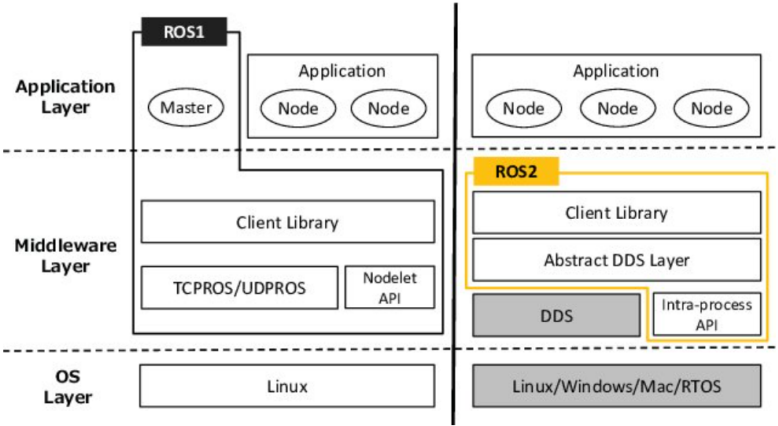
@https://web2.qatar.cmu.edu/~gdicaro/16311-Fall17/slides/Lab-1-ROS-Intro.pdf

ROS1 Big Picture



@https://trojrobert.github.io/hands-on-introdution-to-robot-operating-system(ros)/

ROS2 Big Picture



Y. Maruyama, S. Kato and T. Azumi, "Exploring the performance of ROS2," 2016 International Conference on Embedded Software (EMSOFT), Pittsburgh, PA, USA, 2016, pp. 1-10, doi: 10.1145/2968478.2968502.

ROS2 Installation

1. Option 01: Linux-based users http://wiki.ros.org/melodic/Installation/Ubuntu
2. Option 02: Non Linux-based users. First, you need to install vmware or virtualbox. Second, install Linux-based operating system, e.g., Ubuntu 20.x or Ubuntu 22.x
3. Install docker https://docs.docker.com/engine/install/ubuntu/, and set the user permission https://www.digitalocean.com/community/tutorials/how-to-install-and-use-docker-on-ubuntu-22-04
4. Clone the repository https://github.com/GPrathap/ros2_intro.git

ROS2 Installation

1. Ubuntu 22.0x https://docs.ros.org/en/humble/Installation/Ubuntu-Install-Debians.html