

# Week1

☰ 태그	
☰ 내용	

## 1장 ROS 2 소개

ROS 2는 로봇 개발을 위한 오픈 소스 툴킷으로, 이전 버전인 ROS와 달리 더 많은 기능과 개선된 성능을 제공합니다. ROS 2는 다양한 플랫폼을 지원하며, 로봇 개발자들이 시스템의 개발 및 관리를 쉽게 할 수 있도록 도와줍니다. 또한, ROS 2는 분산 시스템을 지원하며, 높은 신뢰성과 보안성을 제공합니다.

- ROS 는 Robot Operation System 의 약자 입니다.
- ROS 는 전통적인 운영체제가 아닙니다. ROS 는 **메타 운영 체제** 라고 합니다. 기존 운영 체제 위에서 동작하는 로봇 소프트웨어를 위한 미들웨어와 프레임워크 입니다.

### ROS 안에는 어떤 것들이 포함되어 있나요?

- ROS 빌드시스템
- ROS 인터페이스 생성, 관리, 사용을 위한 패키지
- 토픽, 서비스, 액션에 사용되는 인터페이스 패키지
- 통신 미들웨어를 포함하는 RMW 패키지
- 다양한 클라이언트 라이브러리 RCL 패키지
- 로보틱스 어플리케이션 프레임워크
- 시뮬레이션, 시각화 도구 등

### ROS 의 역사와 현 버전은?

- 2007년 5월 미국 스탠퍼드 대학교 연구소의 STAIR (STanford AI Robot) 프로젝트를 위해 개발한 로봇 시스템에서 시작되었습니다.
- 2010년 ROS 1.0 첫 버전이 배포 되었습니다.
- 2015년 ROS 2.0 첫 버전이 배포 되었습니다.
- 2023년 4월 현재 Humble hawksbill 버전을 공식적으로 추천하고 있네요.

# ROS Documentation

This site contains documentation for ROS 1 and ROS 2 distributions. Select your distribution below.

Active ROS 2 distributions

Recommended

Development



## 2장 ROS 2 기반 로봇 개발에 필요한 정보

본문에 정리된 ROS2 레퍼런스 사이트

ROS 커뮤니티 게시판	<a href="https://discourse.ros.org/">https://discourse.ros.org/</a>
ROS 각 버전별 문서	<a href="https://docs.ros.org/">https://docs.ros.org/</a>
ROS2 설계 문서	<a href="https://design.ros2.org/">https://design.ros2.org/</a>
ROS 위키 페이지	<a href="https://wiki.ros.org/">https://wiki.ros.org/</a>
ROS2 Git 저장소	<a href="https://github.com/ros2">https://github.com/ros2</a>

흥미로운 사이트가 하나 있네요. ROS 유저, 오픈소스 사용량등의 통계를 볼 수 있는 사이트입니다.

Users - ROS Metrics

<https://metrics.ros.org/>

## 3장 ROS 2 개발환경 구축

윈도우 WSL2에 **Foxy Fitzroy** 을 설치합니다.

### ROS2 Foxy Fitzroy installation

패키지 시스템 apt 업데이트를 수행하고, curl gnupg2 를 설치 합니다.

```
sudo apt update && sudo apt install curl gnupg2 lsb-release
```

ROS2 apt 저장소를 추가하고 GPG 인증을 수행합니다.

```
sudo curl -sSL https://raw.githubusercontent.com/ros/rosdistro/master/ros.key -o /usr/share/keyrings/ros-archive-keyring.gpg
```

ROS2 저장소를 source.list 에 추가합니다.

```
echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/ros-archive-keyring.gpg] http://packages.ros.org/ros2/ubuntu $(lsb_release -cs) main" | sudo tee /etc/apt/sources.list.d/ros2.list > /dev/null
```

source.list 가 수정되었기 때문에, 패키지 시스템 apt 를 다시 업데이트 합니다.

```
sudo apt update
```

ROS2 기본 패키지인 Communication libraries, message packages, command line tools 과 rvis, demos tutorials 등이 모두 포함된 desktop 버전을 설치합니다.

```
sudo apt install ros-foxy-desktop
```

Bash 터미널에서 환경변수를 로딩 하거나, 아니면 자동으로 로딩 될 수 있도록 .bashrc 에 추가합니다.

```
source /opt/ros/foxy/setup.bash
```

우분투 20.04에 ROS2 Foxy Fitzroy 설치를 완료 하였습니다.

## 4장 왜? ROS 2로 가야하는가?

### ROS2 은 더이상 배포되지 않습니다

ROS1은 2007년 5월에 개발이 시작되어 2020년 5월에 13번째 Noetic Ninjemys 배포판이 마지막으로 출시되었습니다. 반면, ROS2는 2014년 3월부터 개발이 시작되어 2015년에 첫 번째 알파 버전이 배포되었습니다. 2023년 4월 현재 Humble hawksbill 버전을 공식적으로 추천하고 있네요.

### ROS1 의 기술적 한계

1. 임베디드 시스템에 포팅이 어려움
2. 소프트웨어 이식, 확장의 문제
3. 빌드 시스템, 도구의 개선필요
4. 보안문제

## 성능의 문제, 임베디드 포팅이 가능한가?

ROS의 초기 환경은 단일 로봇이며 워크스테이션급 리소스가 필요했습니다. 임베디드 시스템을 고려한 실시간 요구사항이 없었고요. 우수한 네트워크 환경을 전제로 주로 학계에서 연구 응용프로그램을 개발하는 형태였습니다.

로봇 소프트웨어가 다양한 분야로 전파되면서 제한적인 임베디드 환경에 포팅할 필요성이 생겼습니다. 최근 로봇이나 자율주행과 응용분야에서 실시간성이 중요한 이슈로 제기 되었습니다.

## 소프트웨어 이식과 확장의 문제

제한적인 개발 플랫폼

개발 플랫폼이 제한적이고, 보다 다양한 언어의 지원이 필요합니다. 비교적 오래된 오픈소스 라이브러리가 포함되어, 최신 소프트웨어와 연동도 문제가 됩니다.

- Linux, macOS 개발환경
- XMLRPC, TCPROS 을 이용한 통신
- C++03, python2.7

## 빌드 시스템, 도구의 개선필요

초기에는 rosbuidl, catkin (CMake) 등이 통합 빌드 환경을 제공하는 것이 중요한 의미를 가졌습니다. CMake 기반의 catkin은 Python 패키지 관리가 불가능하고, 단일 워크스페이스에서만 사용 가능한 문제가 있었습니다.

- rosbuidl, catkin (CMake-based, catkin\_make, catkin\_tools)
- rosws, wstool, rosininstall

## 보안문제

ROS1은 하나의 마스터(roscore)를 중심으로 만들어졌습니다. 따라서 마스터의 IP, 포트 등이 노출되면 전체 시스템이 위험에 빠질 수 있습니다.