ASMR LAB
Autonomous Systems and Multimodal Reasoning Lab

Introduction to Autonomous Driving: ROS 2 & Docker

Hyeong-jun Kim, Assistant Professor

ASMR Lab. / Department of Future Automotive Engineering

Gyeongsang National University

ROS2 개요

ASMR LAB
Autonomous Systems and Multimodal Reasoning Lab

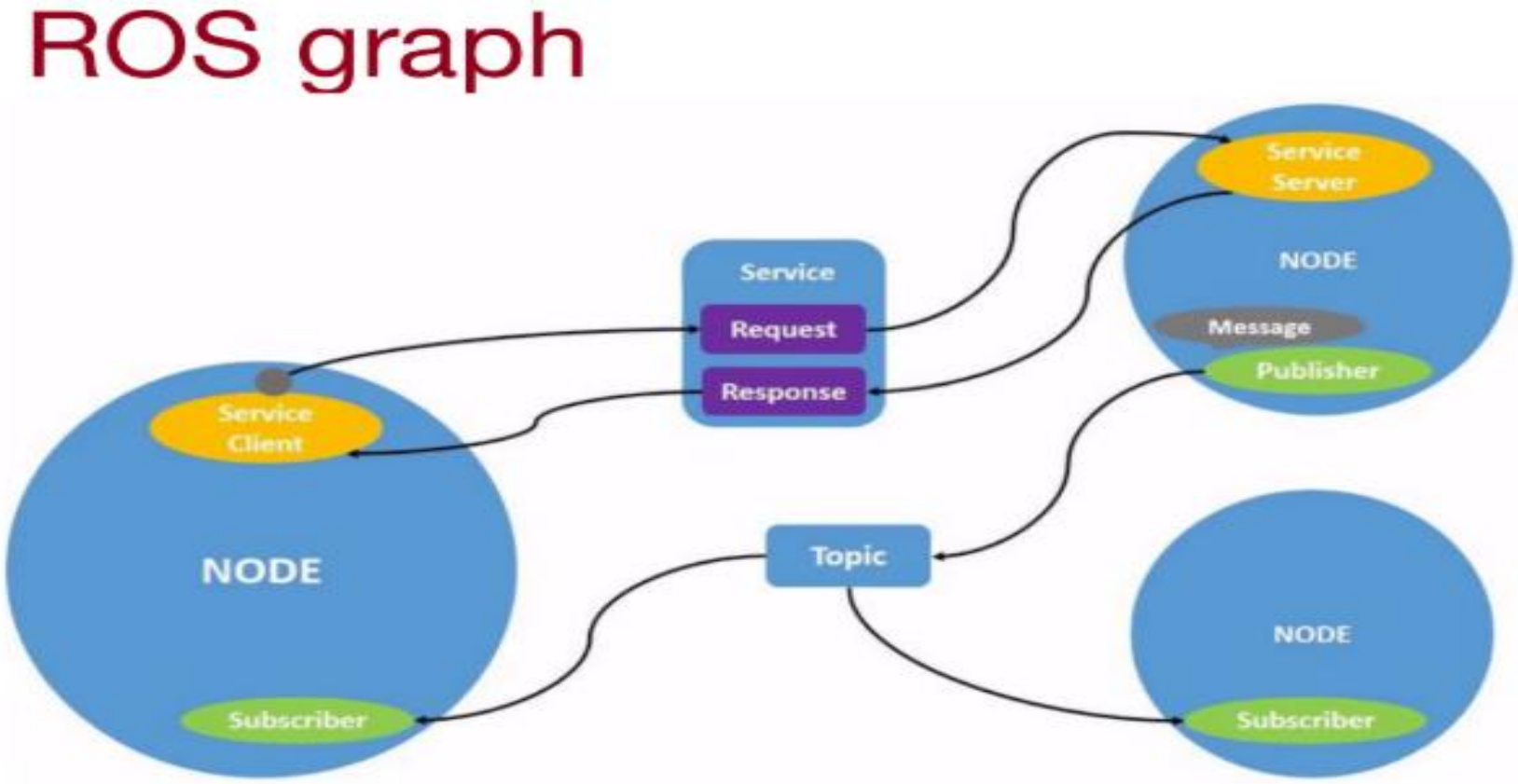
ROS (Robot Operating System)란?:

- 로봇 응용 프로그램 개발을 돕는 로봇 운영 체제
- 핵심 철학: P2P(Peer-to-peer), 분산형(Distributed), 다 언어 지원(Multi-lingual), 경량화(Light-weight), 무료 및 오픈소스(Free and open-source)

ROS 1 vs. ROS 2:

- 언어 표준: C++11 이상 및 Python 3 지원
- 미들웨어: 상용(Off-the-shelf) 미들웨어 사용, DDS(Data Distribution Service)를 통한 탐색, 전송, 직렬화 지원
- 주요 개선: 실시간(Real-time) 기능 지원, Python과의 더 강력한 통합

ROS 그래프 (The ROS Graph)



실행 파일들과 그들 사이의 연결을 시각화한 네트워크로, 모든 데이터 처리가 이루어지는 공간

노드 (Node)

ASMR LAB
Autonomous Systems and Multimodal Reasoning Lab

노드 (Node):

- 바퀴 모터 제어, 센서 처리 등 단일 목적을 가진 실행 프로세스
- 토픽, 서비스 등을 통해 다른 노드와 데이터를 주고받음

관련 명령어 (Related Commands):

- ros2 run <package> <executable>: 특정 패키지 내의 노드(실행 파일)를 실행함
- ros2 node list: 현재 실행 중인 모든 노드의 목록을 출력함
- ros2 node info <node_name>: 특정 노드의 정보(구독/발행 토픽, 서비스 등)를 확인함

통신 방식 1 - 토픽 (Topics)

ASMR LAB
Autonomous Systems and Multimodal Reasoning Lab

개념:

- 복잡한 시스템을 모듈식 노드로 분해하며, 노드들이 메시지를 교환하는 버스(Bus) 역할
- 수행 시스템의 서로 다른 부분 간 데이터를 이동시키는 주요 방법

특징:

- 발행-구독 (Publisher-Subscriber)
- 다대다 통신: 1:1 통신뿐만 아니라 1:N, N:1, N:N 통신 가능
- 노드는 여러 토픽을 동시에 발행하거나 구독 가능

관련 명령어 (Related Commands):

- rqt_graph: 현재 실행 중인 노드와 토픽의 관계를 시각화하여 보여줌
- ros2 topic list: 활성화된 모든 토픽의 목록을 보여줌 (-t 옵션: 메시지 타입 포함)
- ros2 topic echo <topic_name>: 해당 토픽으로 발행되는 메시지 데이터를 터미널에 실시간으로 출력함
- ros2 topic info <topic_name>: 토픽의 메시지 타입, 퍼블리셔/서브스크라이버 수 등 정보를 확인함
- ros2 topic pub <topic> <type> <args>: 터미널에서 직접 토픽에 데이터를 발행함

통신 방식 2 - 서비스 (Services)

ASMR LAB
Autonomous Systems and Multimodal Reasoning Lab

개념:

- 요청(Request)과 응답(Response)으로 이루어지는 양방향 통신

특징:

- 클라이언트가 요청할 때만 데이터를 주고받음 (토픽과의 차이점)
- 하나의 서비스 서버에 여러 클라이언트가 연결될 수 있음

관련 명령어 (Related Commands):

- ros2 service list: 현재 활성화된 모든 서비스 목록을 출력함
- ros2 service type <service_name>: 특정 서비스의 메시지 타입(구조)을 확인함
- ros2 service call <name> <type> <args>: 터미널에서 직접 서비스 요청을 보내고 응답을 받음

통신 방식 3 - 액션 (Actions)

ASMR LAB
Autonomous Systems and Multimodal Reasoning Lab

개념:

- 장시간 실행되는 작업 (Long running tasks)
- 목표(Goal), 피드백(Feedback), 결과(Result)의 세 부분으로 구성

특징:

- 토픽과 서비스를 기반으로 구축됨
- 선점 가능 (Preemptable): 실행 도중에 작업을 취소할 수 있음 (서비스와의 차이점
- 지속적 피드백: 단일 응답만 반환하는 서비스와 달리 지속적인 피드백 스트림 제공
- 구조: 클라이언트-서버 모델 사용 (액션 클라이언트가 목표를 보내면 액션 서버가 수행)

파라미터 (Parameters)

ASMR LAB
Autonomous Systems and Multimodal Reasoning Lab

개념:

- 정의: 노드의 구성 값(Configuration value)으로, 노드의 설정(settings)으로 볼 수 있음
- 데이터 타입: 정수, 실수, 불리언, 문자열, 리스트 등 저장 가능\

특징:

- ROS 2에서 각 노드는 자신의 파라미터를 유지 관리함
- 실행 중에 동적으로 재구성(Dynamically reconfigurable) 가능하며 ROS 2 서비스를 기반으로 함
- 런치(Launch) 파일이나 YAML 파일을 통해 설정 가능

파라미터 관련 명령어:

- ros2 param list: 각 노드별 파라미터 목록을 보여줌
- ros2 param get <node> <param>: 특정 파라미터의 현재 값을 읽어옴
- ros2 param set <node> <param> <value>: 실행 중에 파라미터 값을 변경함

워크스페이스 (Workspace) 및 패키지 구조

ASMR LAB
Autonomous Systems and Multimodal Reasoning Lab

개념:

- ROS 2 패키지들이 포함된 디렉터리

계층 구조 (Layering):

- Underlay: 기본 설치된 ROS 2 환경 (예: /opt/ros/foxy)
- Overlay: 사용자가 작업하는 보조 워크스페이스. Underlay 위에 패키지를 추가하거나 덮어씀(Override)

환경 설정 (Sourcing):

- 터미널 사용 전 설치된 워크스페이스를 source 해야 함
- 주의: .bashrc 파일에 Overlay sourcing 명령어를 넣는 것을 권장하지 않음 (잘못된 Overlay 로드로 인한 오류 방지)

ROS 2 패키지 & 빌드

ASMR LAB
Autonomous Systems and Multimodal Reasoning Lab

패키지 (Package):

- ROS 2 코드를 담는 컨테이너로, 코드 배포 및 공유를 위해 필수적임
- ament 빌드 시스템과 colcon 빌드 도구 사용

패키지 필수 파일 (Python 기준):

- package.xml: 패키지 메타 정보 포함
- setup.py: 패키지 설치 지침 포함
- setup.cfg: 실행 파일 위치 지정 (필수)

빌드 명령어 (Colcon):

- colcon build: 워크스페이스 루트에서 실행(워크 스페이스 내의 모든 패키지를 빌드함)
- --packages-up-to: 특정 패키지와 의존성만 빌드
- --symlink-install: Python 스크립트 수정 시 재 빌드 과정을 생략해 줌
- rosdep install ...: 패키지의 의존성 라이브러리를 자동으로 설치함
- source install/setup.bash: 빌드 된 환경을 현재 터미널에 적용함

Docker 도입 (Introduction to Docker)

ASMR LAB
Autonomous Systems and Multimodal Reasoning Lab

Docker 개요:

- 소프트웨어를 느슨하게 격리된 샌드박스(컨테이너) 환경에서 배포할 수 있게 해주는 도구
- 애플리케이션과 모든 의존성을 표준화된 단위로 패키징 하여 호스트 설치 환경에 의존하지 않음

VM vs Container:

- VM: 게스트 OS를 포함하여 무겁고 비효율적
- Container: 호스트 OS의 커널을 공유하며 프로세스만 격리하므로 가볍고 빠름

Docker 핵심 개념

ASMR LAB
Autonomous Systems and Multimodal Reasoning Lab

이미지 (Image):

- 컨테이너를 생성하기 위한 읽기 전용 템플릿
- Dockerfile의 각 명령어가 레이어(Layer)를 생성하며, 변경된 레이어만 재 빌드됨

컨테이너 (Container):

- 이미지의 실행 가능한 인스턴스
- 호스트 머신과 격리되어 있으며, 삭제 시 저장되지 않은 상태는 사라짐

Dockerfile:

- 이미지를 자동으로 빌드하기 위한 명령어 집합
- 주요 명령어: FROM(베이스 이미지), RUN(명령어 실행), COPY(파일 복사), CMD/ENTRYPOINT(실행 설정)

Docker 주요 명령어 (Basic Commands)

ASMR LAB
Autonomous Systems and Multimodal Reasoning Lab

이미지 및 컨테이너 관리:

- docker pull <image>: 레지스트리(Hub)에서 이미지를 다운로드함
- docker images: 다운로드된 이미지 목록을 보여줌
- docker ps: 현재 실행 중인 컨테이너 목록을 보여줌 (-a: 정지된 것도 포함)
- docker build: Dockerfile을 기반으로 새로운 이미지를 생성함

실행 및 제어:

- docker run [options] <image>: 이미지를 기반으로 새 컨테이너를 생성하고 시작함
- docker exec -it <container> bash: 실행 중인 컨테이너 내부에 접속하여 명령어를 입력함
- docker rm <container>: 컨테이너를 삭제함 docker rmi <image>: 이미지를 삭제함

Docker 데이터 관리 (Bind Mounts)

ASMR LAB
Autonomous Systems and Multimodal Reasoning Lab

Bind Mount의 필요성:

- 컨테이너 내부 데이터의 휘발성 문제 해결 및 개발 편의성 증대

기능:

- 호스트 머신의 파일이나 디렉터리를 컨테이너 내부의 경로에 직접 마운트함
- 호스트의 절대 경로를 참조하여 연결

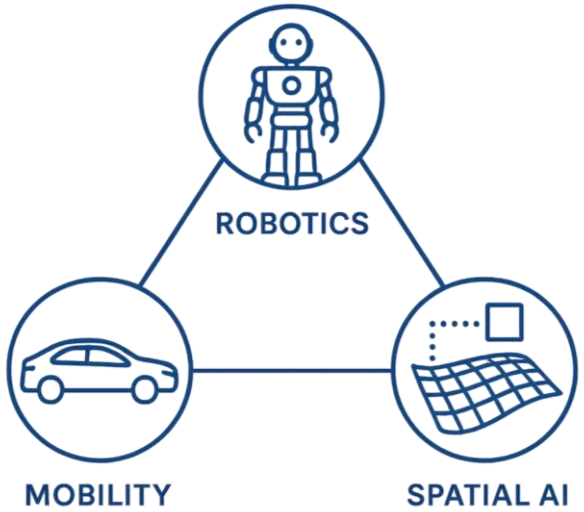
사용 명령어:

- docker run -v <host_path>:<container_path> ...
- 예: -v $(pwd):/sim_ws/src (현재 폴더를 컨테이너의 소스 폴더에 연결)
- 개발 시 호스트에서 코드를 수정하면 컨테이너에 즉시 반영됨.

ASMR LAB
Autonomous Systems and Multimodal Reasoning Lab

Thank you for your attention

Questions and feedback are welcome.



ASMR Lab.
Autonomous Systems and Multimodal Reasoning Lab.

hj.kim@gnu.ac.kr