

- 소프트웨어(Software)

- 정보들이 이동하는 방향과 정보 처리의 종류를 지정
- 한 동작들이 일어나는 시간을 지정해주는 명령의 집합
- 사람이 이해하기 쉬운 고급 언어로 작성
- 시스템 소프트웨어 : OS(win, linux, IOS 등), 링커, 컴파일러 등
- 응용 소프트웨어 : 워드프로세서, 웹 브라우저, 한글 등



시스템 소프트웨어 (System Software)

각종 유틸리티 프로그램
(링커, 로더, 컴파일러 등등)

운영체제(Operating System)

셸(shell)

커널(kernel)

운영체제(Operating System)

- 하드웨어 / 모든 소프트웨어 관리하는 '관리자 프로그램'

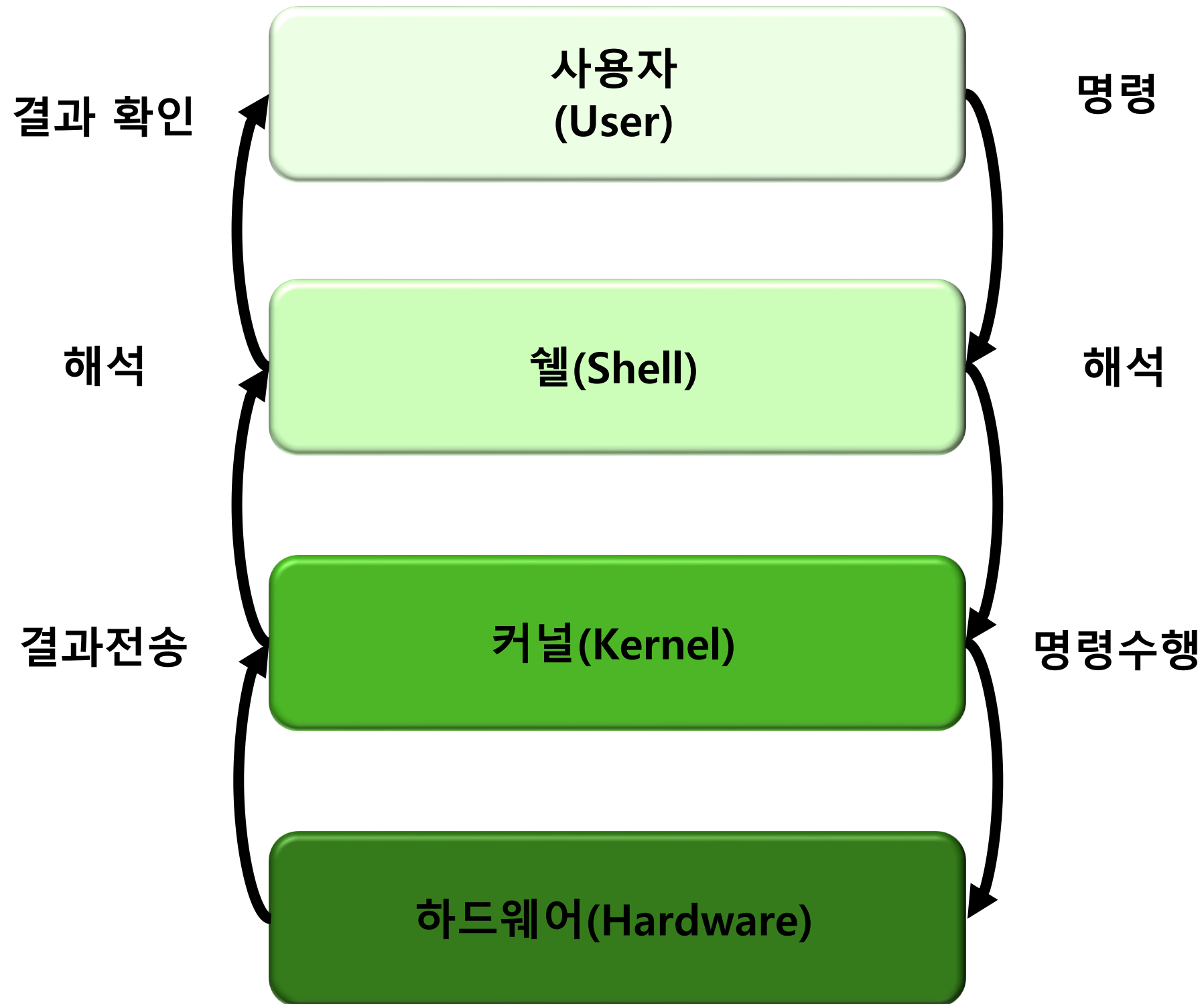
사용자 영역

사용자가 응용프로그램을
이용하기 위해서 마련한 공간

커널 영역

운영체제의 핵심적 역할
CPU, 메모리, 입출력장치 등과
같은 중요한 자원 초기화 및
관리

컴퓨터 소프트웨어 - 운영체제



셸(Shell)

- 운영체제 상에서 다양한 운영체제 기능과 서비스를 구현하는 인터페이스 제공 프로그램
- 명령어 해석기로 사용자와 OS간의 인터페이스 해석
- 키보드와 같은 단말 장치 통해 유저 입력 받아서 실행
- Sh, ash, bash, csh 등 다양한 셸 존재
- Cmd 창도 셸의 일종

커널(Kernel)

- 컴퓨터 운영체제의 가장 중요한 핵심
- 커널은 부팅 될 때 메모리로 로딩됨
- 로딩화면이 커널 로딩 화면
- 컴퓨터의 시스템 자원 관리
- 리눅스 커널은 10000줄 이상의 C언어와 1000줄 정도의 어셈블리어로 구성되어 있어 이식성 좋음

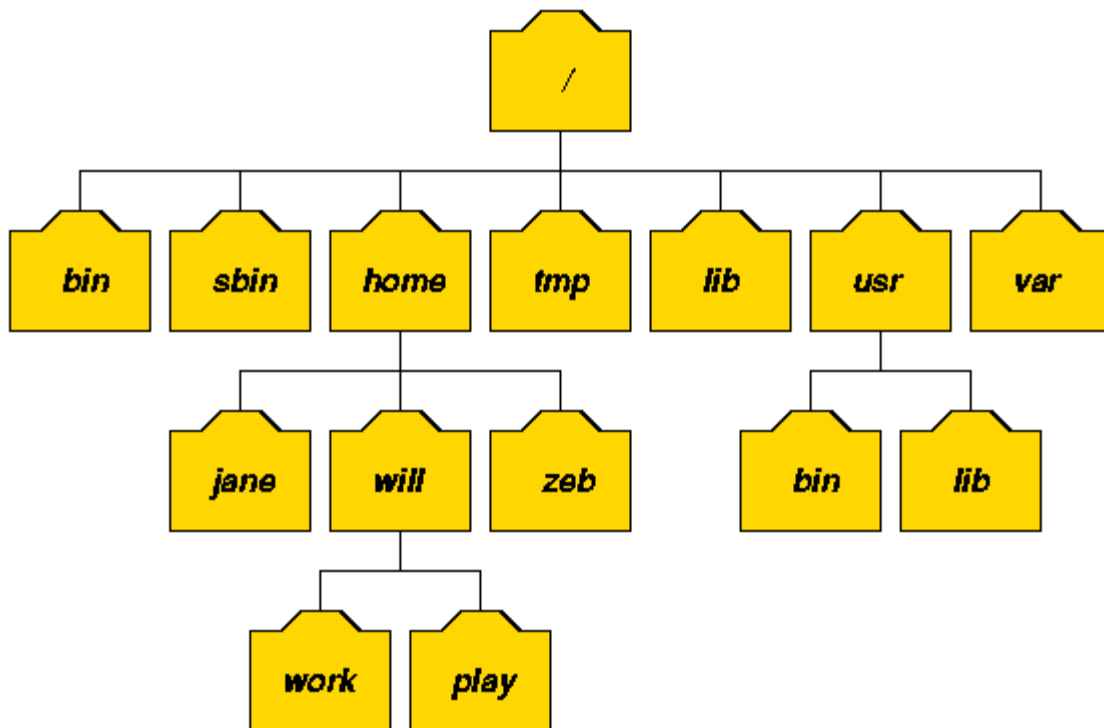
리눅스 개념



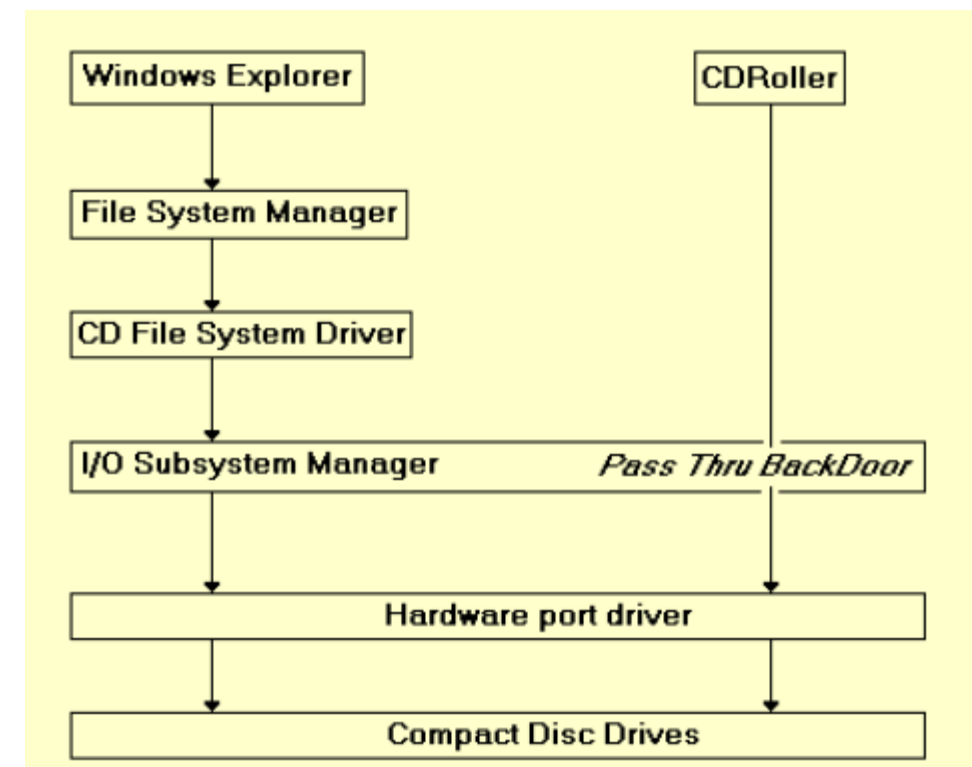
- 리누스 토발즈(Linus Torvalds)가 만든 UNIX 클론
- Open source의 운영체제
- Linux는 전세계 개발자들이 자발적으로 참여하고 만들어 가는 운영체제
- 현재도 활발히 개발되고 있음
- 현재 전 세계적으로 300여 가지의 배포판 존재
- 대표적 배포판 : redhat, Ubuntu, Suse, Debian, Gentoo

파일 시스템

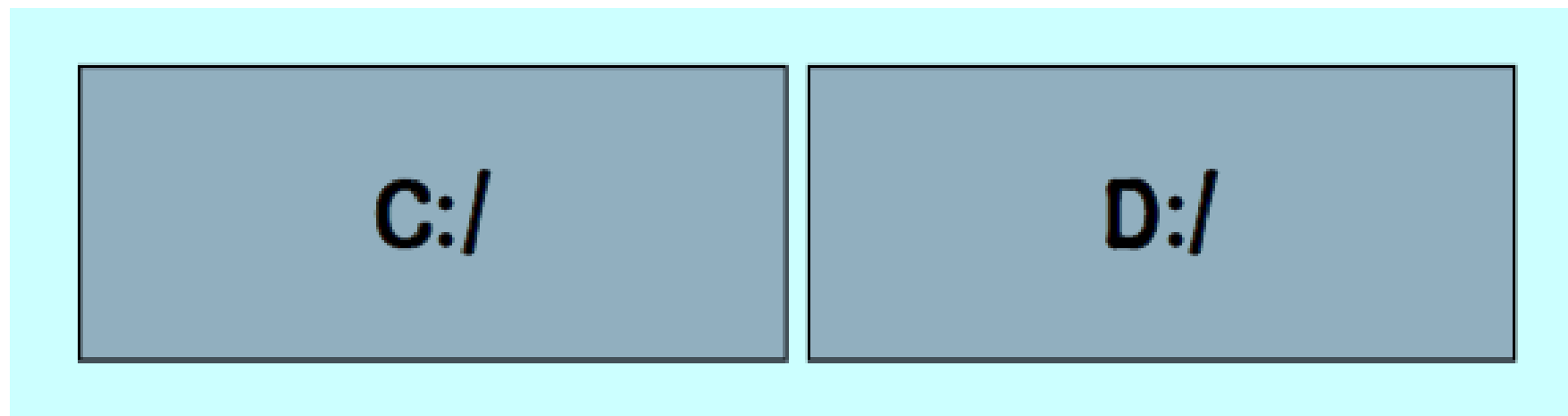
- 컴퓨터 파일들을 저장하고 체계적으로 관리하는 방법
- 파일이름으로 파일들 관리
- Metadata(data(파일)에 관한 data)를 이용하여 파일의 정보 기록
- 디렉토리(폴더)들로 파일을 계층적으로 나누어 분류
- 종류 : NTFS, FAT, EXT, HFS, ...



리눅스 디렉토리 구조 (그림 출처 <http://www.doc.ic.ac.uk/~wjk/UnixIntro/Lecture2.html>)



- Windows : C:₩, D:₩라는 파티션(partition)구분

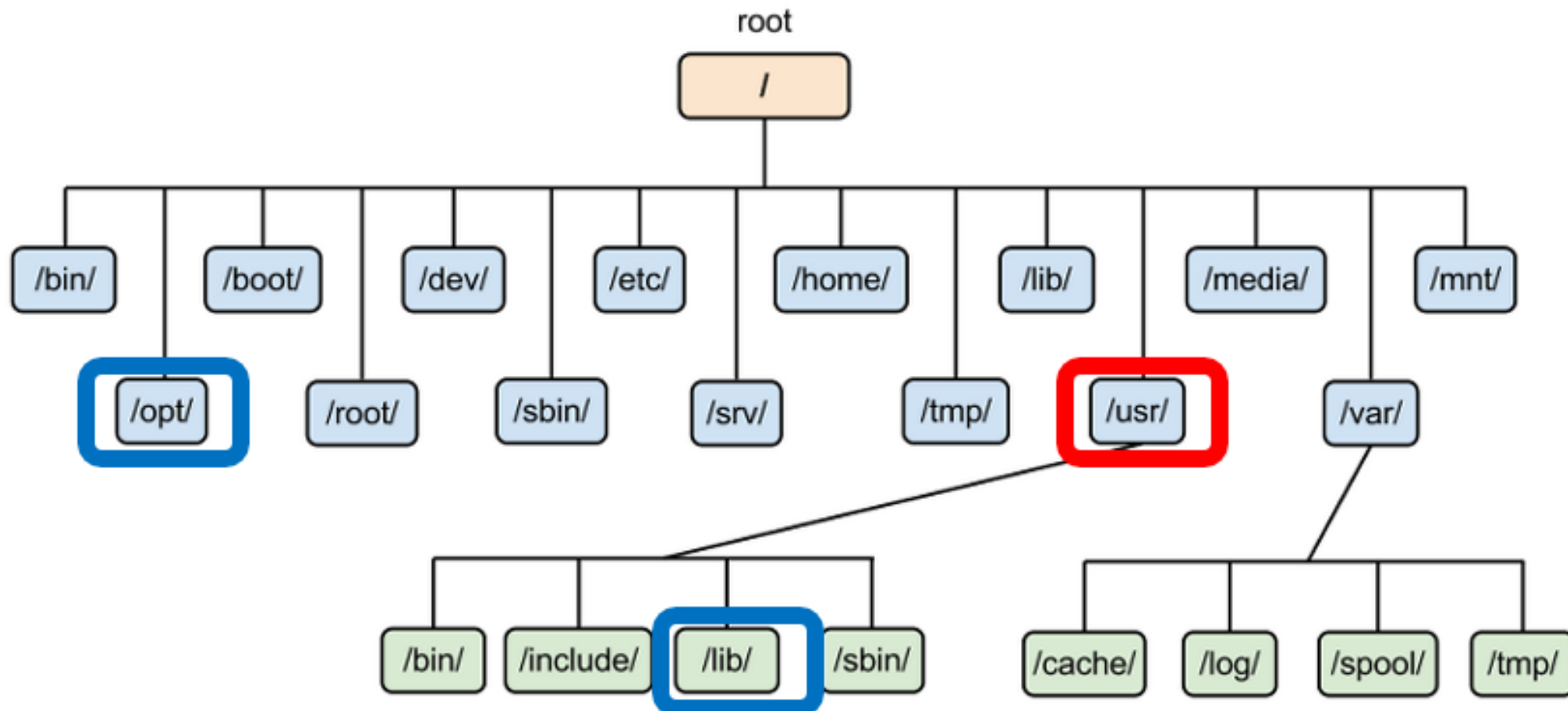


파일 시스템 - 리눅스

- 리눅스에서 관리하는 모든 것은 파일로 인식
- '디렉토리'이용하여 전체 파일을 하나의 트리로 만듦
- 트리의 root는 /로 나타내며 root directory라고 함
- '디렉토리' : 하위의 디렉토리들과 파일들의 정보 저장하는 파일
- Absolute path : /를 기준으로 한 경로
- Relative path : /를 포함하지 않는 경로, 쉘의 작업 디렉토리가 상대 경로의 기준
- 디렉토리 자기자신을 '.', 바로 위의 디렉토리는 '..'으로 표현

파일 시스템 - 리눅스

- Absolute path (절대 경로) : /를 기준으로 한 경로
- Relative path : /를 포함하지 않는 경로, 쉘의 작업 디렉토리가 상대 경로의 기준
- 디렉토리 자기자신을 '.', 바로 위의 디렉토리는 '..'으로 표현



사용자 계층

- root : 최고 관리자, 컴퓨터와 사용자들 관리
- User : root외의 일반 사용자들
- Soder : sudo라는 유틸리티에 의해 정의된 사용자
 - ➔ sudo는 root가 허가한 사용자에게 대해 일정시간(보통 5분) 동안 일부 명령을 root로 사용
 - ➔ /etc/sudoers 파일에 sudo를 사용할 수 있는 사용자와 각 사용자가 root권한으로 실행할 수 있는 명령어 정의
- 권한을 변경할 수 있는 권한은 파일의 소유주와 root
 - ➔ Read/Write/Execute

R	W	X	R	W	X	R	W	X
User			Group			Other		

디렉토리 구조

/bin /sbin /boot /tmp /dev

/usr /etc /var /home /lib

/media /proc /opt /root

/home, /root

- /home은 일반 사용자들의 홈 디렉토리
- /root는 루트의 홈 디렉토리

/bin, /sbin

- /bin은 일반사용자가 리눅스 사용할 때 쓰는 각종 유틸리티 들어있는 디렉토리 (cat, chmod, ls ..)
- /sbin은 루투유저가 시스템 관리에 필요한 유틸리티가 들어있는 디렉토리

/boot, /dev

- /boot는 부팅에 필요한 필수적인 파일들이 모인 디렉토리
- /dev는 장치 파일, 특수 파일이 들어있는 폴더(모든 하드웨어를 파일로 인식)

/etc, /lib

- /etc은 시스템의 전반적인 관리 파일이 들어있는 디렉토리(설정 파일)
- /lib는 자주 사용되는 라이브러리를 모아놓음

/media, /proc

- /media는 외부 저장매체를 마운트 하여 하위 디렉토리인 것 같이 사용
- /proc은 RAM에 존재하며

실행되고 있는 프로세스, 사용중 디바이스, 커널이 수집한 하드웨어 정보 저장

/opt, /tmp

- /opt는 기본적으로 설치되는 프로그램을 제외한
소프트웨어들로 주로 규모가 있는 소프트웨어 패키지 들어있음
- /tmp는 임시파일들이 저장되는 장소로 linux 종료 시 모든 파일 삭제

/media, /proc

- /media는 외부 저장매체를 마운트 하여 하위 디렉토리인 것 같이 사용
- /proc은 RAM에 존재하며

실행되고 있는 프로세스, 사용중 디바이스, 커널이 수집한 하드웨어 정보 저장

/opt, /tmp

- /opt는 기본적으로 설치되는 프로그램을 제외한
소프트웨어들로 주로 규모가 있는 소프트웨어 패키지 들어있음
- /tmp는 임시파일들이 저장되는 장소로 linux 종료 시 모든 파일 삭제

/usr

- /usr는 시스템에 빌드하는 프로그램들의 소스나 매뉴얼 등을 포함하는 디렉토리
- 가장 큰 공간 차지

/var

- 자주 바뀌는 데이터들을 모아놓은 디렉토리

pwd (print working directory)

- 현재 작업중인 디렉토리 표시

cd (change directory)

- 작업중인 디렉토리 다른 디렉토리로 변경

ls (list)

- 파일들을 나열하는 명령어

-l(long)

- 권한, group, owner 등 자세한 정보 표시

-a(all)

- 숨겨진 파일까지 다 보여짐

-R(recursive)

- 디렉토리 안의 디렉토리 안의 ... 디렉토리 까지 전부 나온다.

-d(directory)

- 디렉토리 내부 콘텐츠가 아니라 디렉토리 그 자체의 정보를 출력한다.

- ls -d public_html vs ls public_html

passwd (password)

- 비밀번호 변경
- passwd → 기존 비밀번호 → 새 비밀번호

chsh (change shell)

- 사용하는 shell을 바꾸어 줌

man (mannual)

- 프로그램/명령어 사용방법 출력
- man [command]

cp (copy)

- 파일 복사

-r(recursive) : 해당 파일과 그 하위파일까지 전부 복사
• 디렉토리 복사 시 사용 가능.

mv (move)

- 파일을 이동함

mkdir (make directory)

- 디렉토리를 새로 만듦

rm (remove)

- 디렉토리 지울 때 사용

rmdir (remove directory)

- 비어있는 디렉토리 지울 때 사용

ROS의 시작

ROS is an open-source, meta-operating system for your robot. It provides the services you would expect from an operating system, including hardware abstraction, low-level device control, implementation of commonly-used functionality, message-passing between processes, and package management. It also provides tools and libraries for obtaining, building, writing, and running code across multiple computers.

<http://www.ros.org/wiki/>



ROS의 시작

- **ROS** = Robot Operating System
- **ROS**는 **메타운영체제**(Meta-Operating System)이다.

로보틱스 소프트웨어 개발을 전 세계 레벨에서
공동 작업 가능하도록 생태계를 구축하는 것!



메타 운영 체제

- **메타운영체제(Meta-Operating System)** 딱히 정확히 정의된 용어는 아니지만, 어플리케이션과 분산 컴퓨팅 자원간의 가상화 레이어로 분산 컴퓨팅 자원을 활용하여, 스케줄링 및 로드, 감시, 에러 처리 등을 실행하는 시스템이라고 볼 수 있다.
- 즉, 윈도우, 리눅스, 안드로이드와 같은 **전통적인 운영체제는 아니다**. 오히려, ROS는 기존의 **전통적인 운영체제(리눅스, 윈도우즈, OS-X, 안드로이드)를 이용**하고 있다.
- 기존 운영체제의 프로세스 관리 시스템, 파일 시스템, 유저 인터페이스, 프로그램 유틸(컴파일러, 스레드 모델 등)등을 사용하고 있다. 이에 추가적으로 다수의 이기종 하드웨어간의 데이터 송수신, 스케줄링, 에러 처리 등 **로봇 응용 소프트웨어 개발을 위한** 필수 기능들을 라이브러리 형태로 제공하고 있다.
- 또한, 이러한 **로봇 소프트웨어 프레임워크**를 기반으로 다양한 목적의 응용 프로그램을 개발, 관리, 제공하고 있으며 유저들이 개발한 패키지 또한 유통하는 **생태계(ecosystem)**를 갖추고 있다.

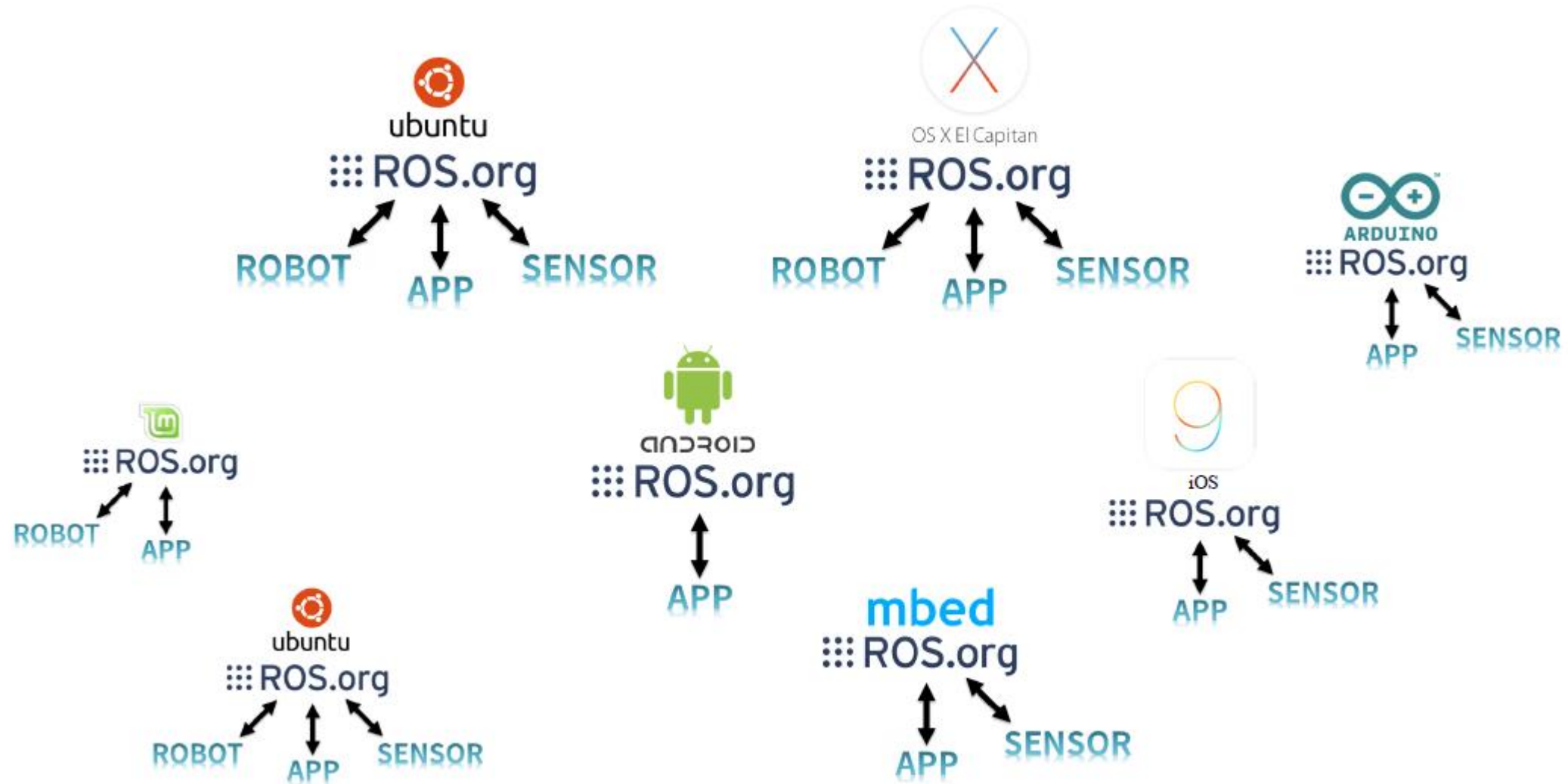
메타 운영 체제



디바이스 드라이버, 라이브러리, 디버그 도구, 메시지 통신
구동 도구, 컴파일 도구, 인스톨러, 패키지 생성 및 릴리즈



메타 운영 체제를 통한 디바이스 간의 통신 지원

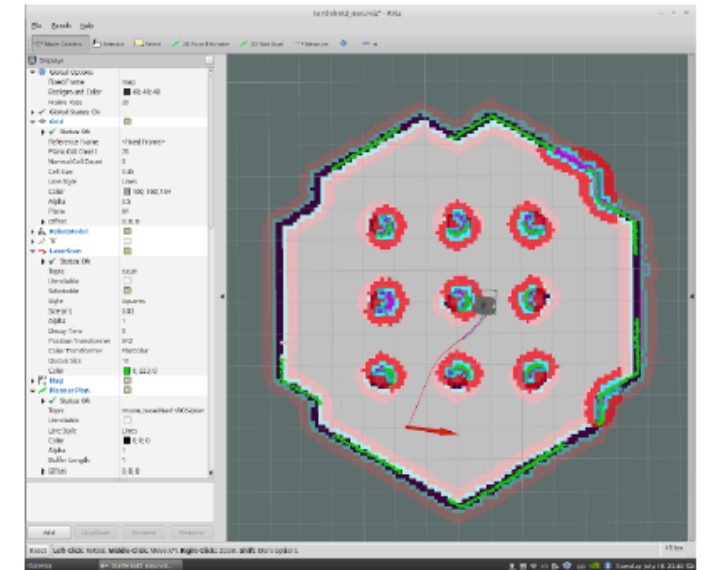
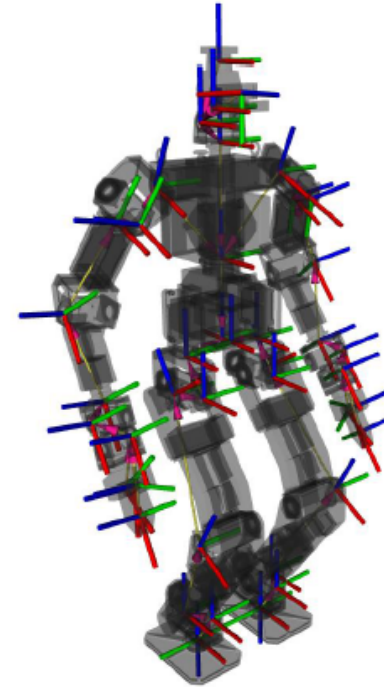


통신 인프라

- 노드 간 데이터 통신을 제공
- 통상적 미들웨어로 지칭되는 메시지 전달 인터페이스 지원
- 메시지 파싱 기능
 - 로봇 개발 시에 빈번히 사용되는 통신 시스템 제공
 - 캡슐화 및 코드 재사용을 촉진하는 노드들 간의 메시지 전달 인터페이스
- 메시지의 기록 및 재생
 - 노드 간 송/수신되는 데이터인 메시지를 저장하고 필요시에 재사용 가능
 - 저장된 메시지를 기반으로 반복적인 실험 가능, 알고리즘 개발에 용이함
- 메시지 사용으로 인한 다양한 프로그래밍 언어 사용 가능
 - 노드 간의 데이터 교환이 메시지를 사용하기 때문에 각 노드는 서로 다른 언어로 작성 가능
 - 클라이언트 라이브러리: roscpp, rospy, roslisp, rosjava, roslua, roscs, roseus, PhaROS, rosR
- 분산 매개 변수 시스템
 - 시스템에서 사용되는 변수를 글로벌 키값으로 작성하여 공유 및 수정하여 실시간으로 반영

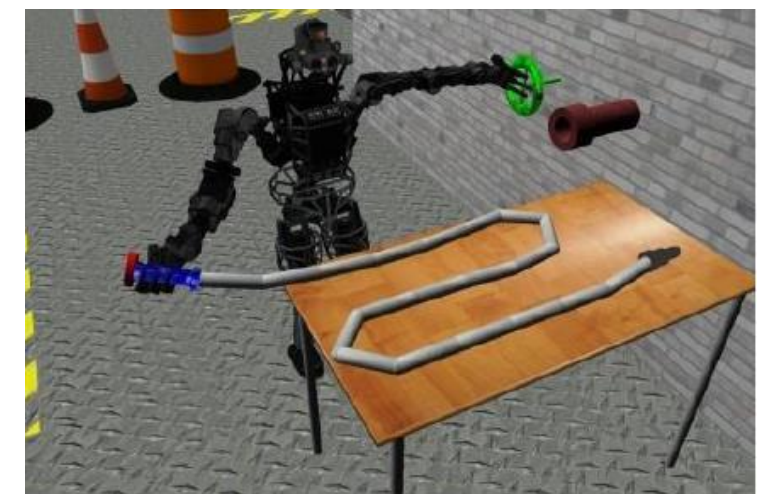
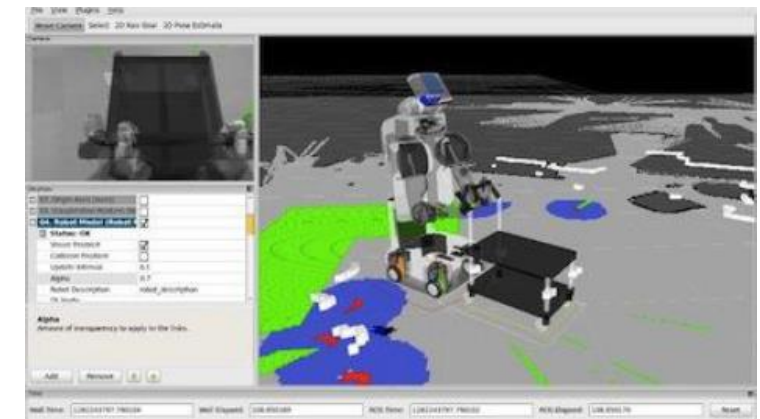
로봇 관련 다양한 기능

- **로봇에 대한 표준 메시지 정의**
 - 카메라, IMU, 레이저 등의 센서 / 오도메트리, 경로 및 지도 등의 내비게이션 데이터 등의 표준 메시지를 정의하여 모듈화, 협업 작업을 유도, 효율성 향상
- **로봇 기하학 라이브러리**
 - 로봇, 센서 등의 상대적 좌표를 트리화 시키는 TF 제공
- **로봇 기술 언어**
 - 로봇의 물리적 특성을 설명하는 XML 문서 기술
- **진단 시스템**
 - 로봇의 상태를 한눈에 파악할 수 있는 진단 시스템 제공
- **센싱/인식**
 - 센서 드라이버, 센싱/인식 레벨의 라이브러리 제공
- **내비게이션**
 - 로봇에서 많이 사용되는 로봇의 포즈(위치/자세) 추정, 지도내의 자기 위치 추정 제공
 - 지도 작성에 필요한 SLAM, 작성된 지도 내에서 목적지를 찾아가는 Navigation 라이브러리를 제공
- **매니플레이션**
 - 로봇 암에 사용되는 IK, FK 는 물론 응용단의 Pick and Place 를 지원하는 다양한 Manipulation 라이브러리 제공
 - GUI 형태의 매니플레이션 Tools 제공(MoveIt!)



다양한 개발 도구

- 로봇 개발에 필요한 다양한 개발 도구를 제공
- 로봇 개발의 효율성 향상
- **Command-Line Tools**
 - GUI 없이 ROS에서 제공되는 명령어로만 로봇 액세스 및 거의 모든 ROS 기능 소화
- **RViz**
 - 강력한 3D 시각화툴 제공
 - 레이저, 카메라 등의 센서 데이터를 시각화
 - 로봇 외형과 계획된 동작을 표현
- **RQT**
 - 그래픽 인터페이스 개발을 위한 Qt 기반 프레임 워크 제공
 - 노드와 그들 사이의 연결 정보 표시(rqt_graph)
 - 인코더, 전압, 또는 시간이 지남에 따라 변화하는 숫자를 플로팅(rqt_plot)
 - 데이터를 메시지 형태로 기록하고 재생(rqt_bag)
- **Gazebo**
 - 물리 엔진을 탑재, 로봇, 센서, 환경 모델 등을 지원, 3차원 시뮬레이터
 - ROS와의 높은 호환성



Linux & ROS 버전 선택

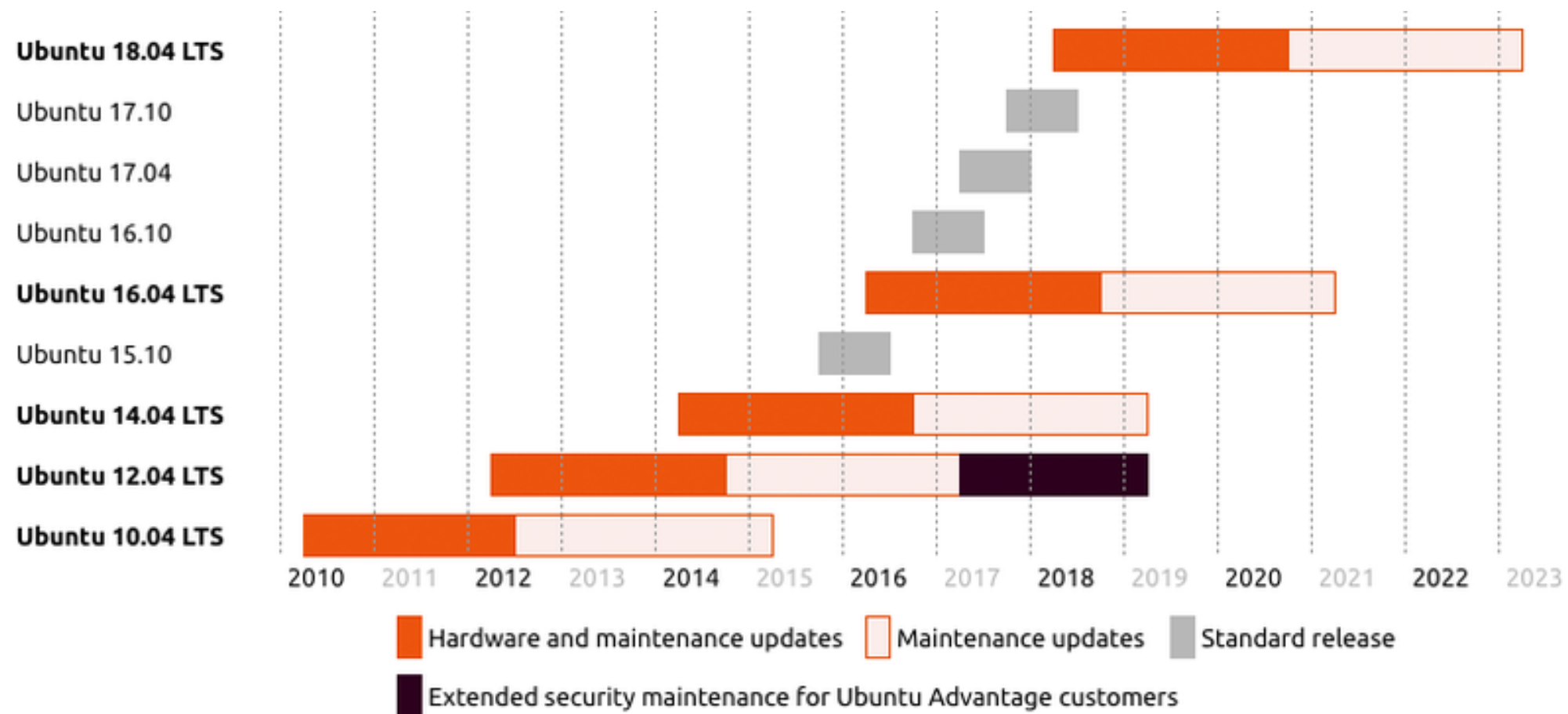
ROS의 역사

Groovy Galapagos	December 31, 2012		2014-07-31
Fuerte Turtle	April 23, 2012		--
Electric Emys	August 30, 2011		--
Diamondback	March 2, 2011		--
C Turtle	August 2, 2010		--
Box Turtle	March 2, 2010		--

Old version
Older version, still maintained
Latest version
Future release

Distribution	Release date	Poster	EOL date
Noetic Ninjemys (last ROS 1 release)	May 23, 2020		May 2025
Melodic Morenia	May 23, 2018		2023-05-30
Lunar Loggerhead	May 23, 2017		2019-05-30
Kinetic Kame	May 23, 2016		2021-05-30
Jade Turtle	May 23, 2015		2017-05-30
Indigo Igloo	July 22, 2014		2019-04-30
Hydro Medusa	September 4, 2013		2014-05-31

linux의 version 역사



ROS 개발환경 구축

<http://wiki.ros.org/>

ROS.org

[About](#) | [Support](#) | [Discussion Forum](#) | [Service Status](#) | [Q&A answers.ros.org](#)

Search:

Documentation

Browse Software

News

Download

Documentation

ROS (Robot Operating System) provides libraries and tools to help software developers create robot applications. It provides hardware abstraction, device drivers, libraries, visualizers, message-passing, package management, and more. ROS is licensed under an open source, BSD license.

Available Translations: [German](#) | [Spanish](#) | [French](#) | [Italian](#) | [Japanese](#) | [Korean](#) | [Brazilian Portuguese](#) | [Portuguese](#) | [Русский \(Russian\)](#) | [Thai](#) | [Turkish](#) | [简体中文](#) | [Ukrainian](#) | [Vietnamese](#)

ROS:

Introduction

An introduction to what is ROS.

Install

Install ROS on your machine.

Getting Started

Learn about various concepts, client libraries, and technical overview of ROS.

Tutorials

Step-by-step instructions for learning ROS hands-on

Contribute

How to get involved with the ROS community, such as submitting your own repository.

Support

What to do if something doesn't work as expected.

Quality Assurance

How to ensure that your ROS-based systems and your contributions to ROS are of high quality.

Software:

Distributions

View the different release Distributions for ROS.

Packages

Search the 2000+ software libraries available for ROS.

Core Libraries

APIs by language and topic.

Common Tools

Common tools for developing and debugging ROS software.

위키

[Distributions](#)
[ROS/Installation](#)
[ROS/Tutorials](#)
[RecentChanges](#)
[Documentation](#)

문서

[못 고치는 문서](#)
[정보](#)
[첨부](#)
다른 작업:

사용자

[로그인](#)

ROS 기본 용어

■ Node

- 최소 단위의 실행 가능한 프로세서를 가리키는 용어로서 하나의 실행 가능한 프로그램으로 생각하면 된다. ROS 에서는 최소한의 실행단위로 프로그램을 나누어 작업하게 된다. 각 노드는 메시지 통신으로 데이터를 주고 받는다.

■ Package

- 하나 이상의 노드, 노드 실행을 위한 정보 등을 묶어 놓은 것. 또한, 패키지의 묶음을 메타패키지라 하여 따로 분리한다.

■ Message

- 메시지를 통해 노드간의 데이터를 주고받게 된다. 메시지는 integer, floating point, boolean 와 같은 변수형태이다. 또한, 메시지 안에 메시지를 품고 있는 간단한 데이터 구조 및 메시지들의 배열과 같은 구조도 사용할 수 있다.

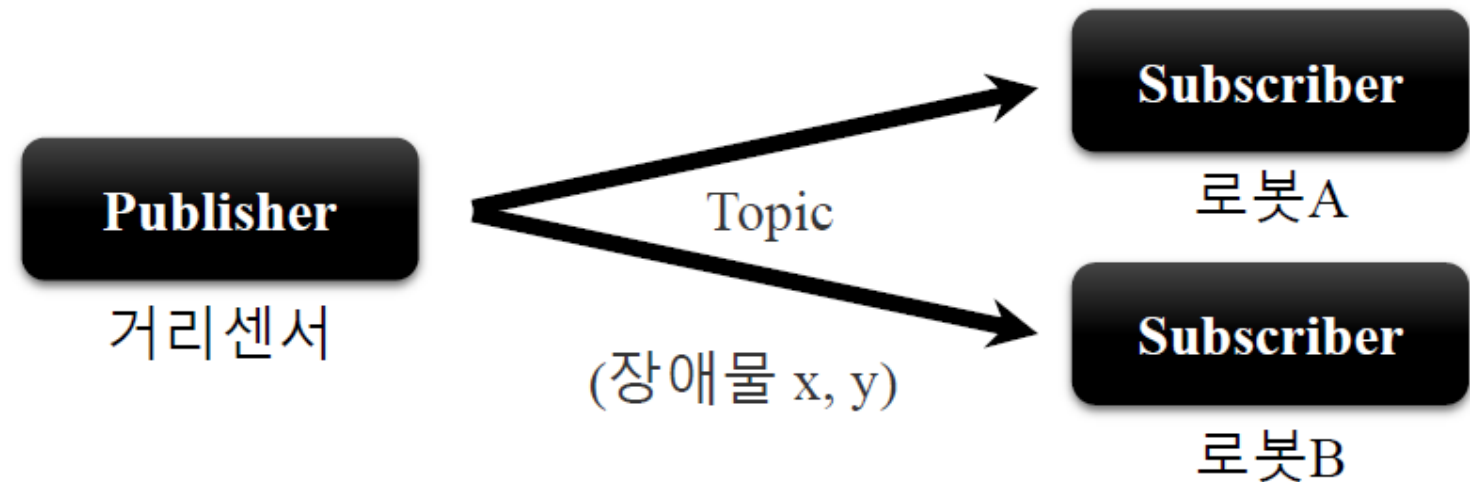
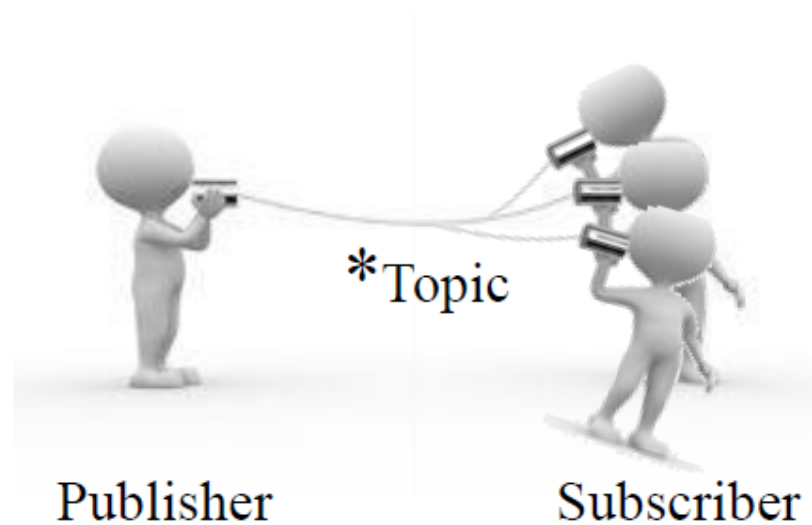
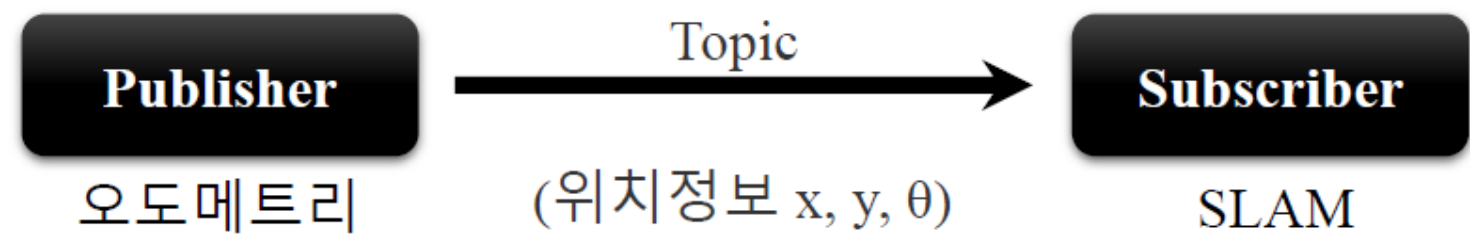
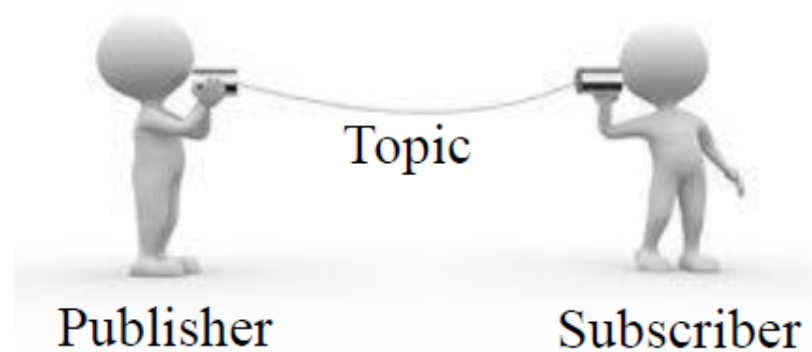
ROS 기본 용어

- ROS
 - ➔ 로봇 응용 프로그램의 개발을 위한 운영체제와 같은 로봇 소프트웨어 플랫폼
- 마스터(Master)
 - ➔ 노드와 노드 사이 연결, 메시지 통신을 위한 네임 서버와 같은 역할, roscore가 마스터 실행 명령이며, 각 노드들의 이름을 등록하고 필요에 따라 정보를 받을 수 있다.
 - 마스터 없이는 노드간의 접속, 토픽과 서비스와 같은 메시지 통신 할 수 없다.

ROS 기본 용어

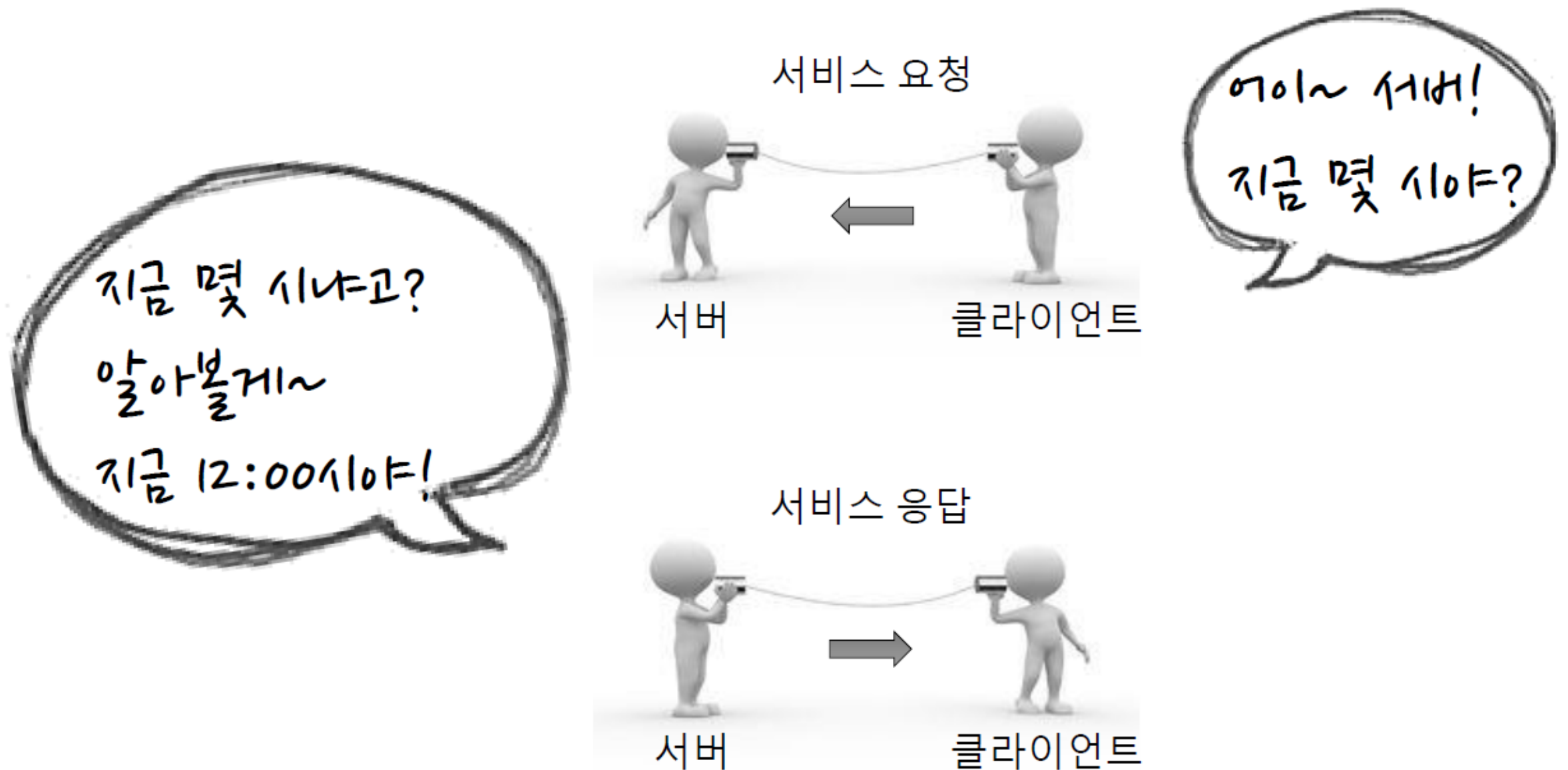
- ROS
- 마스터 (Master)
- 노드 (node)
- 패키지 (package)
- 메타패키지 (metapackage)
- 메시지 (message, msg)
- 토픽 (topic)
- 발행 (publish) 및 발행자 (publisher)
- 구독 (subscribe) 및 구독자 (subscriber)
- 서비스 (service)
- 서비스 서버 (service server)
- 서비스 클라이언트 (service client)
- 케킨 (catkin)
- 로스빌드 (roscbuild)
- 로스코어 (roscscore)
- 매개변수 (parameter)
- 매개변수 서버 (parameter server)
- 로스런 (rosrun)
- 로스런치 (roslaunch)
- 백, 배그 (bag)
- RPC (Remote Procedure Call)
- XML (Extensible Markup Language)
- XMLRPC
- CMakeLists.txt
- Package.sml

Topic, Publisher, Subscriber

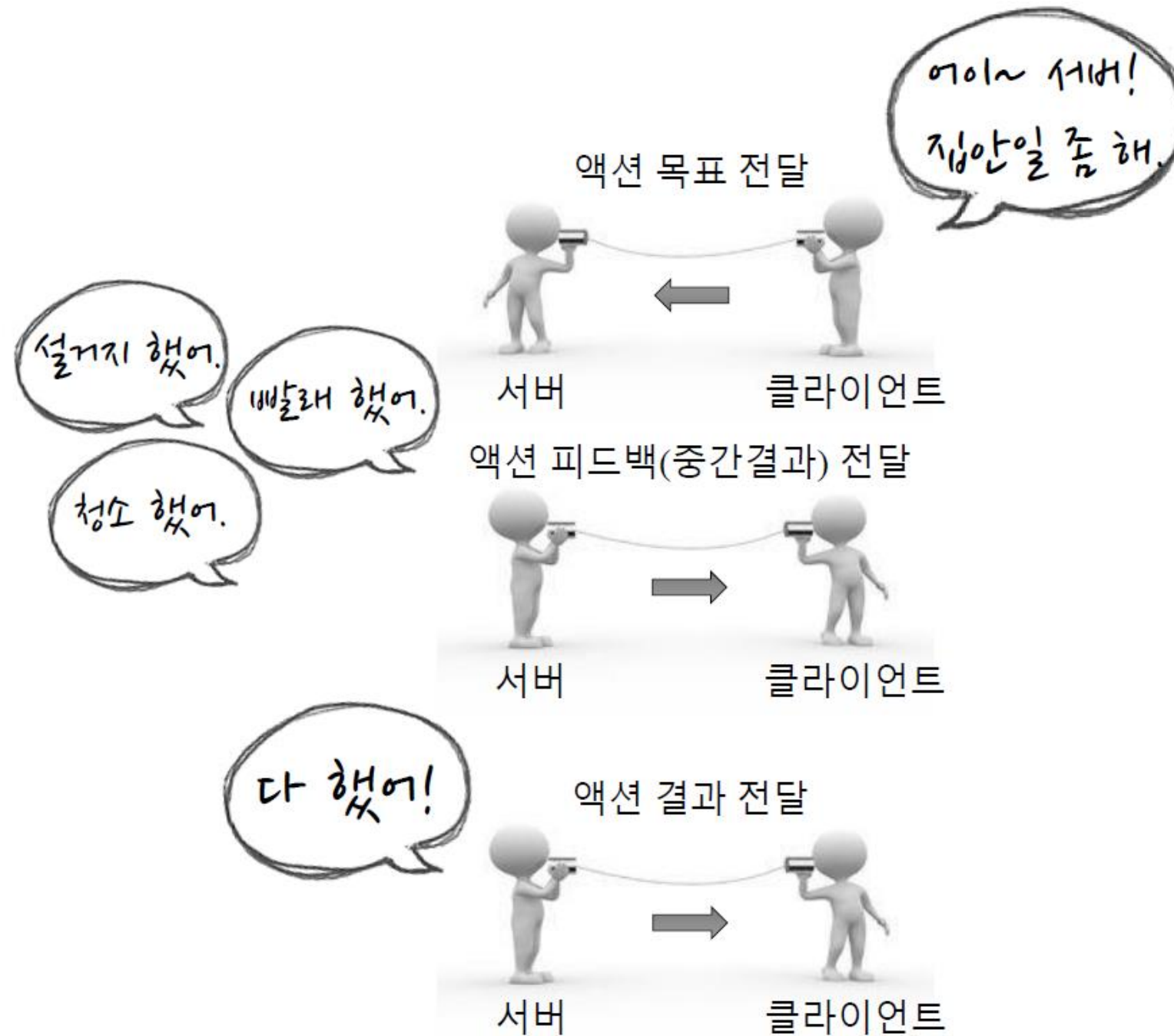


*Topic 에 대해 1:1의 Publisher, Subscriber 통신도 가능하며, 목적에 따라서 1:N, N:1, N:N 통신도 가능하다.

Service, Service server, Service client



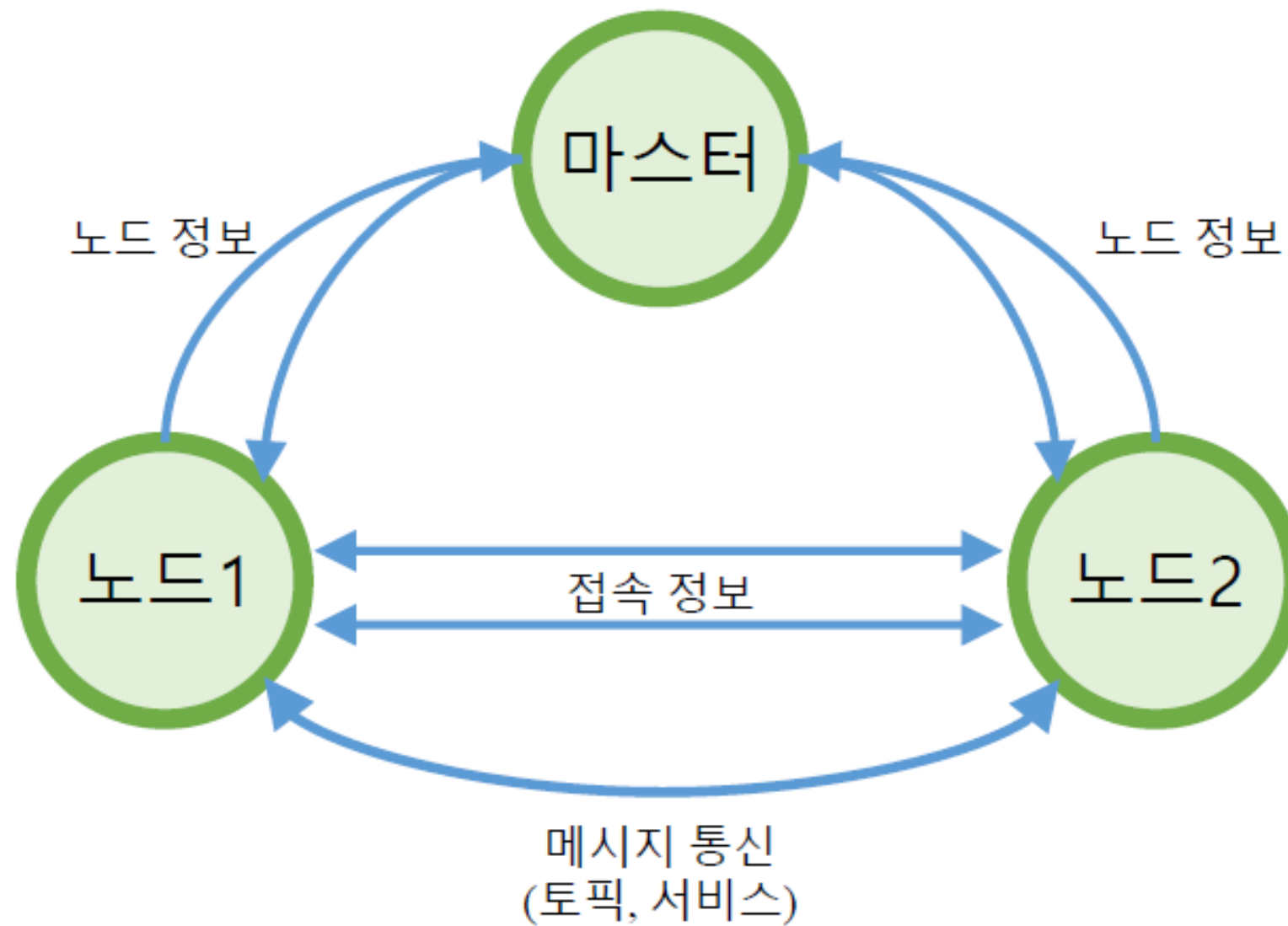
Action, Action server, Action client



정리

종 류	동기 / 비동기	방향	참고
Topic	비동기	단방향	연속적으로 데이터를 송수신하는 경우
Service	동기	양방향	요청 처리가 순간적인 현재 상태의 요청 및 응답
Action	비동기	양방향	요청 처리 후 응답까지 오래 걸리거나 중간 피드백이 필요한 경우

ROS에서 가장 기본 구조



마스터 구동 : XMLRPC

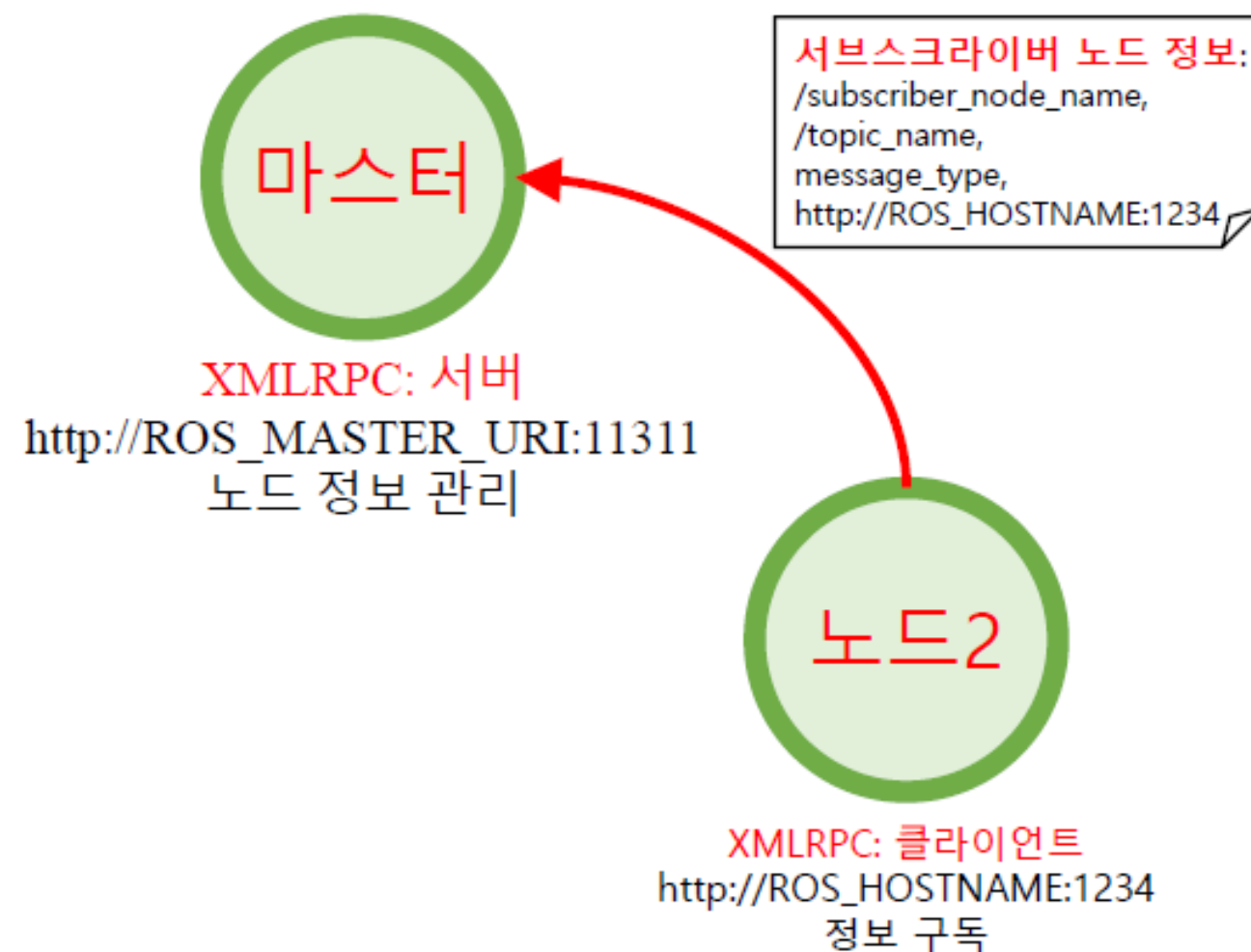
`$ roscore`



XMLRPC: 서버
http://ROS_MASTER_URI:11311
노드 정보 관리

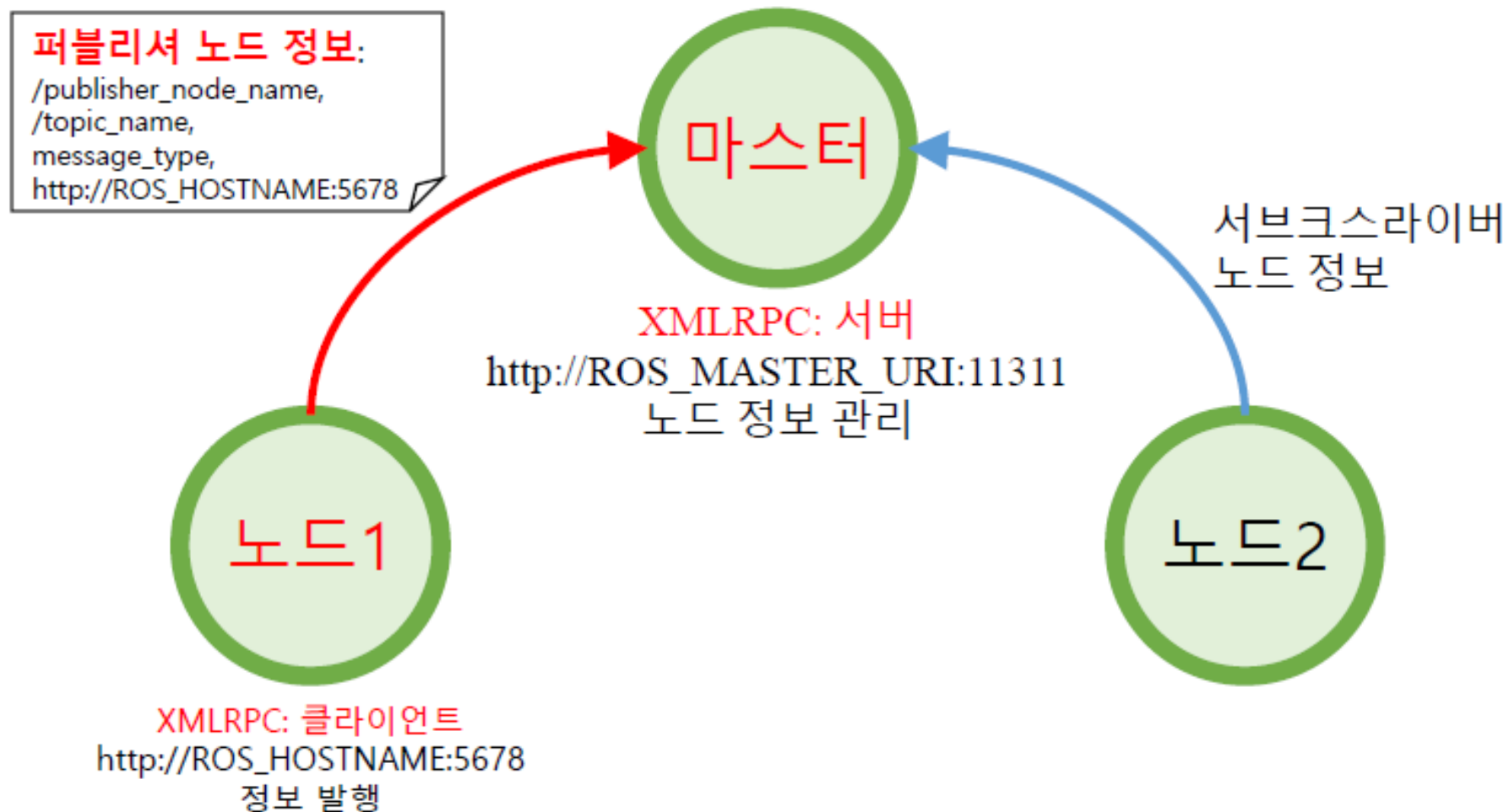
Subscriber node 구동

\$ rosrun 패키지이름 노드이름



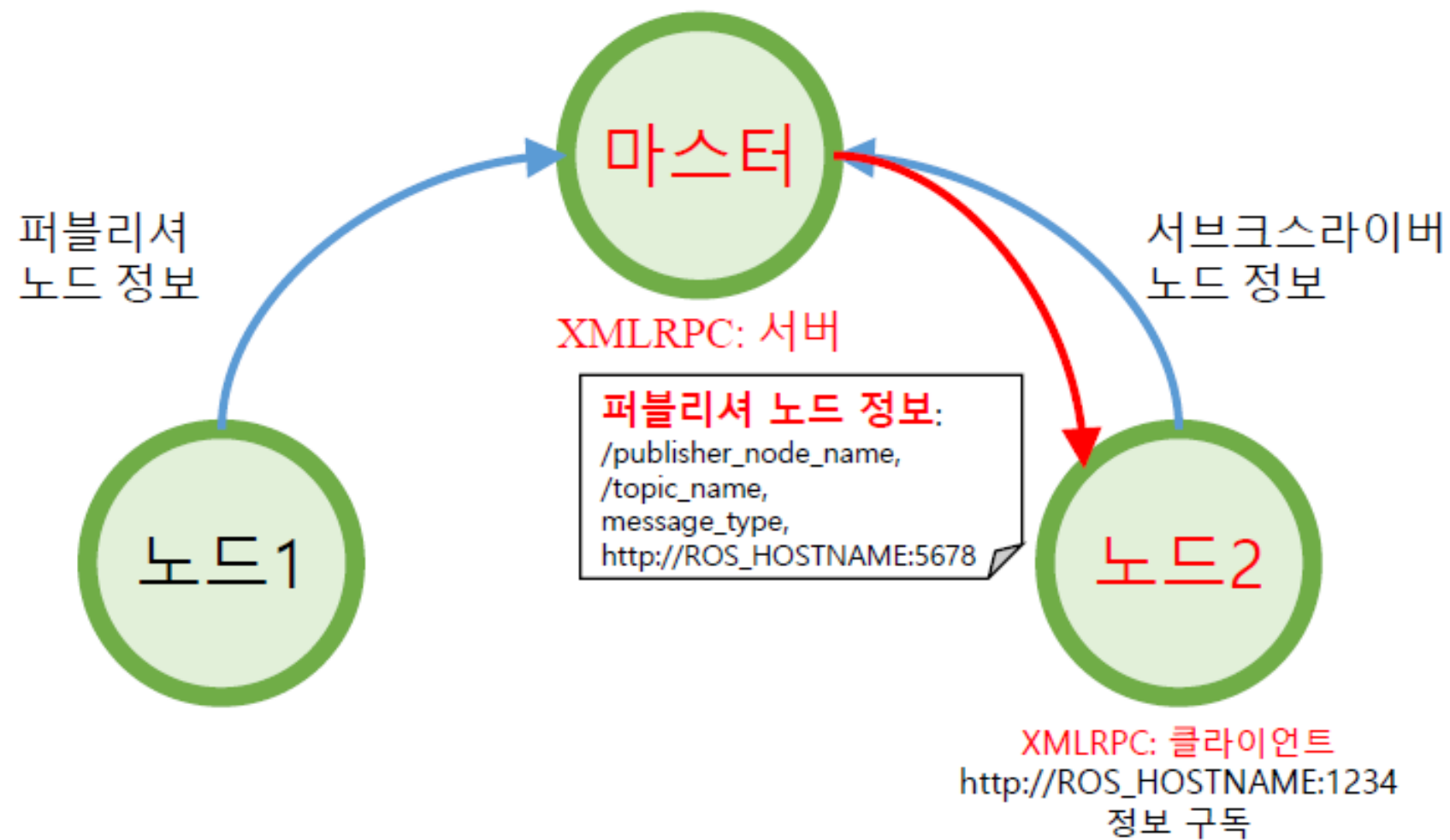
Publisher node 구동

\$ rosrun 패키지이름 노드이름



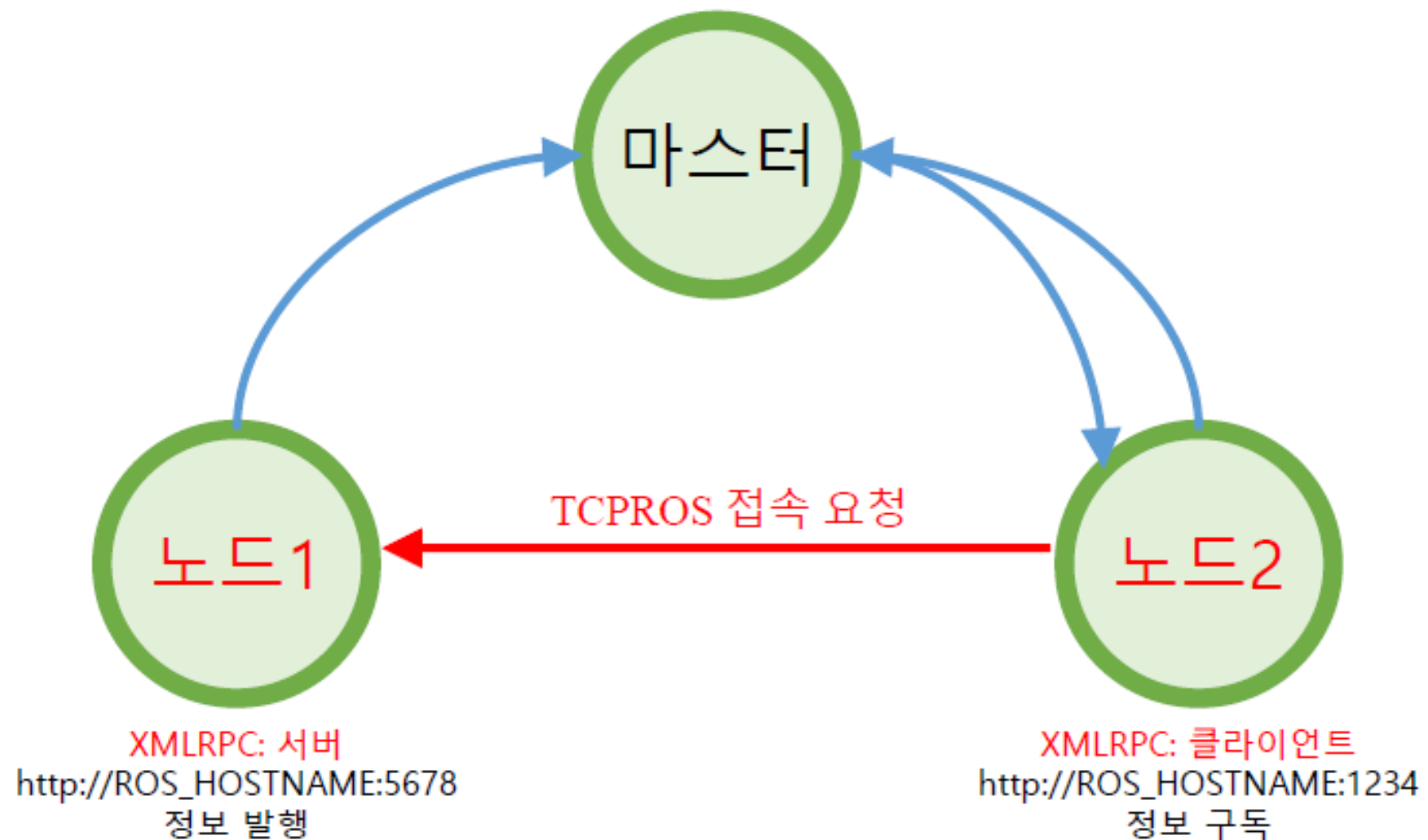
Publisher 정보 알림

➔ 마스터는 subscriber 노드에게 새로운 publisher 정보를 알림



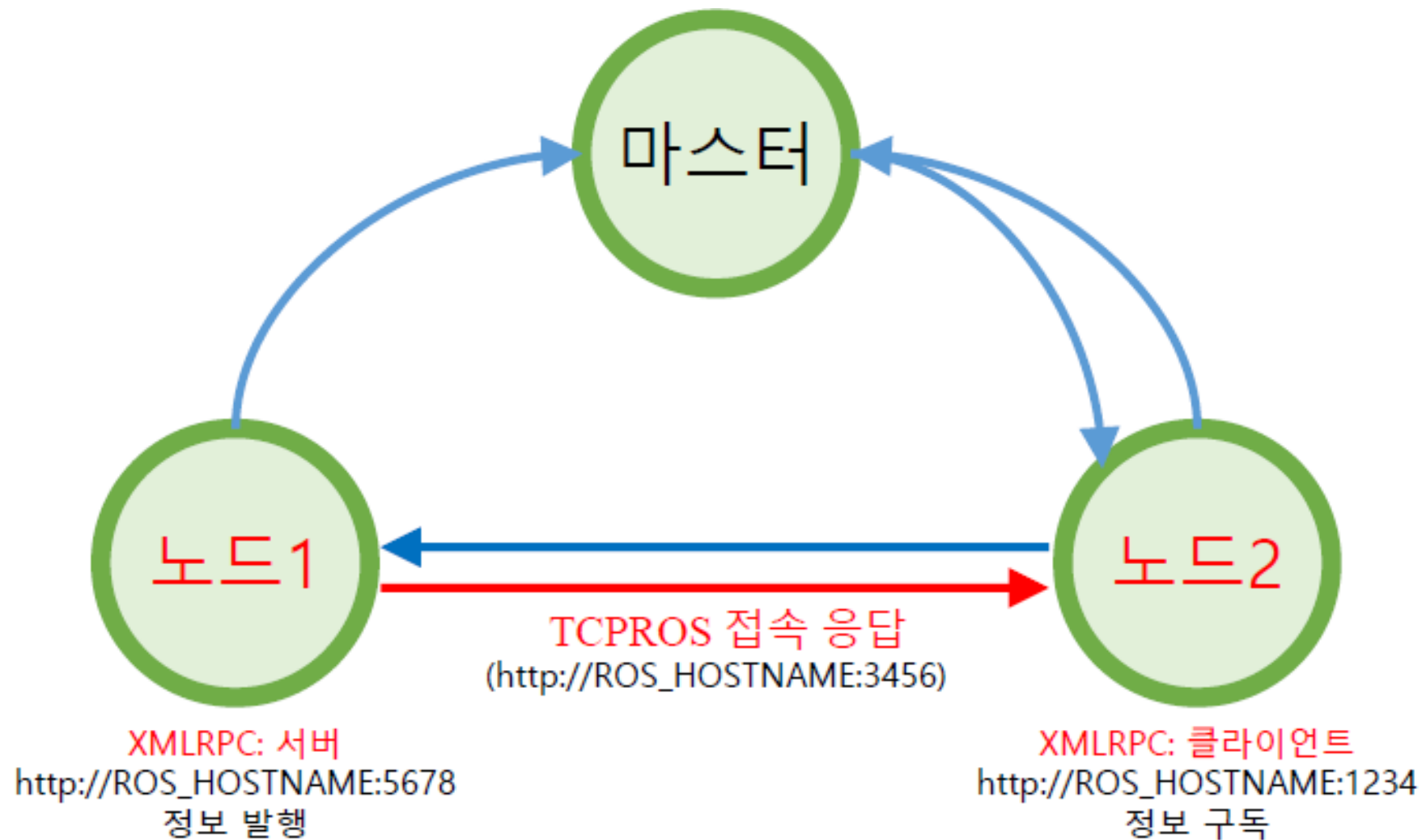
Publisher 노드에 접속 요청

➔ 마스터로부터 받은 publisher 정보를 이용하여 TCPROS 접속 요청



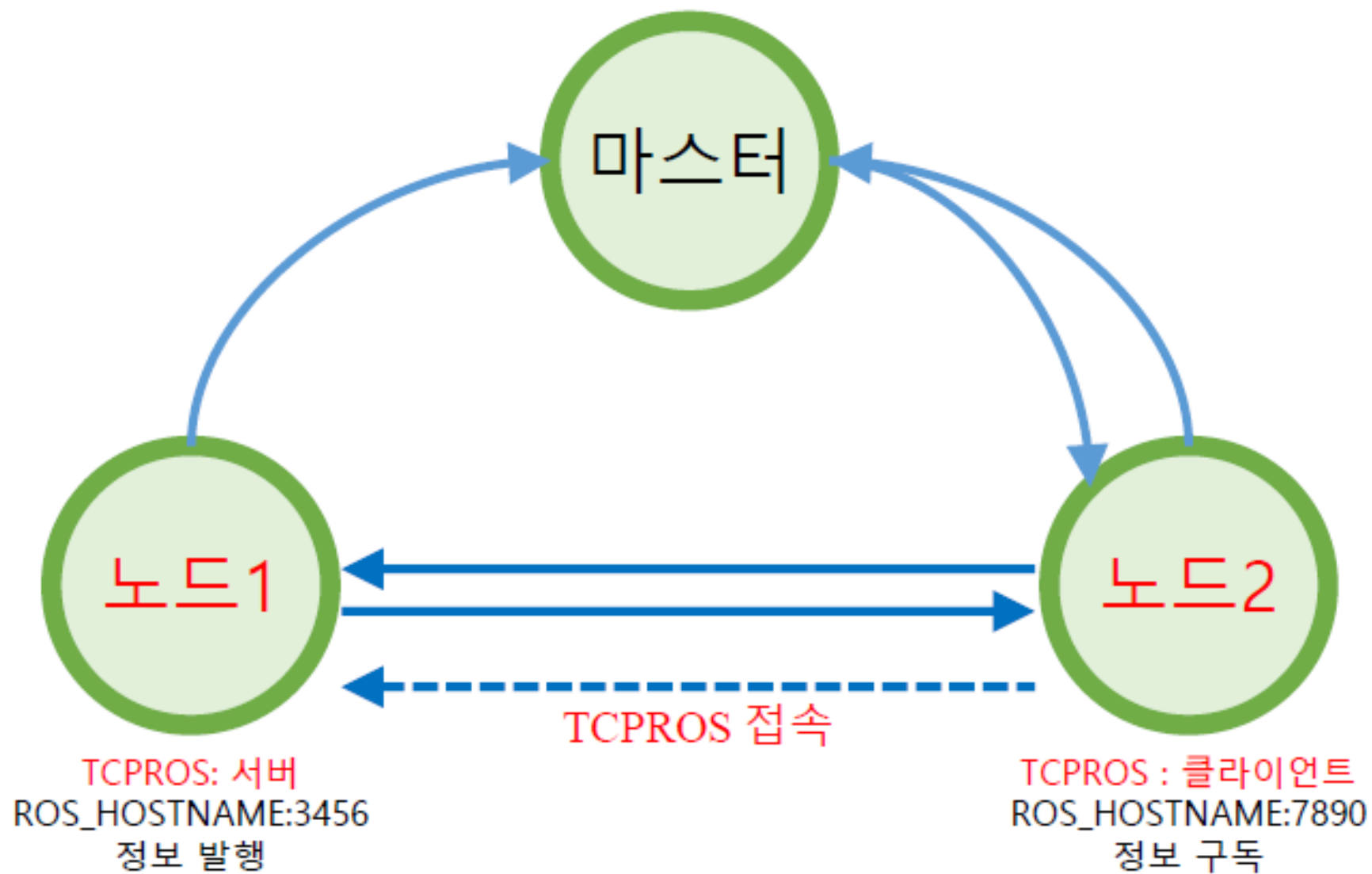
Subscriber 노드에 접속 응답

➔ 접속 응답에 해당되는 자신의 TCP URI 주소와 포트번호 전송



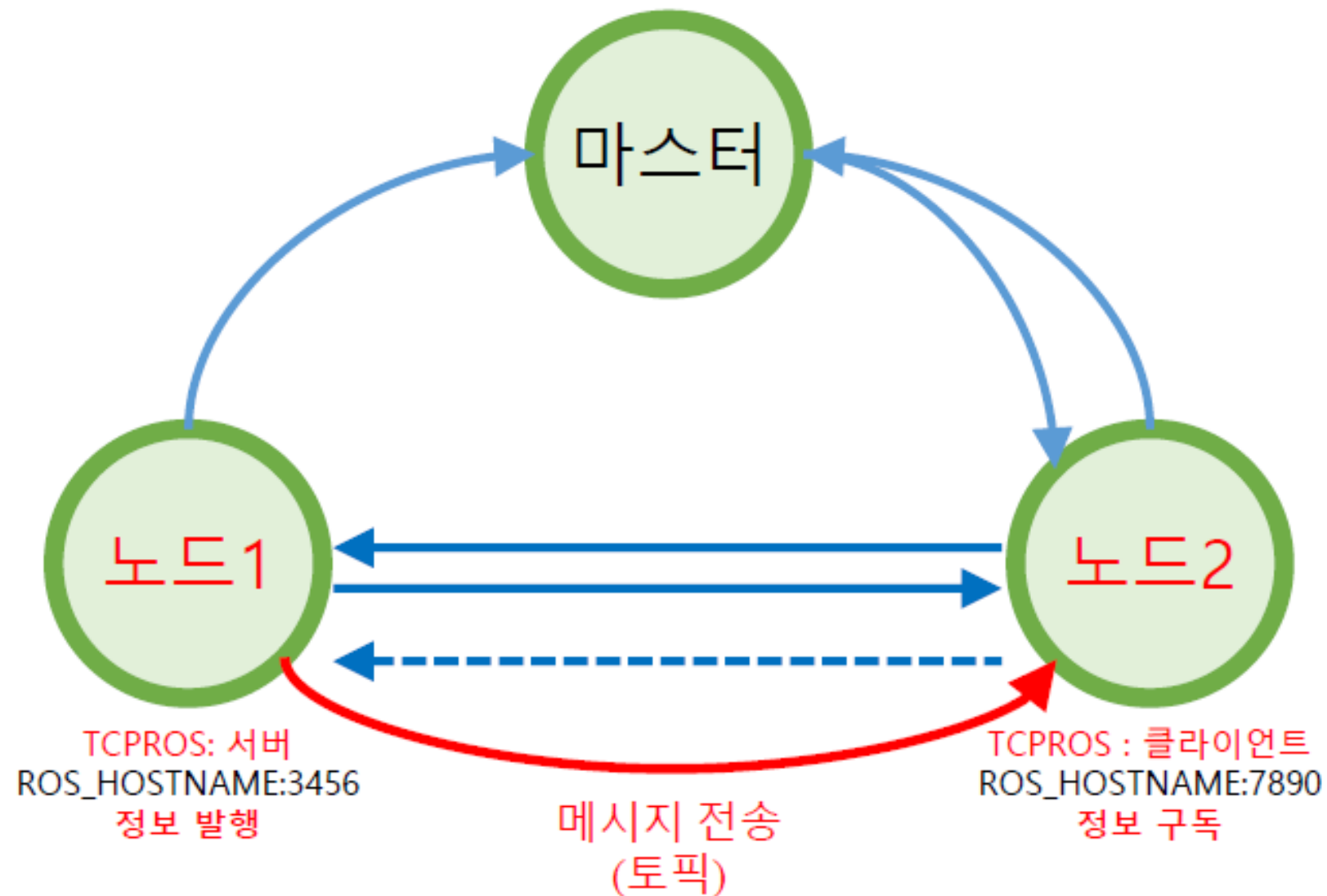
TCP 접속

➔ TCPROS를 이용하여 publisher 노드와 직접 연결



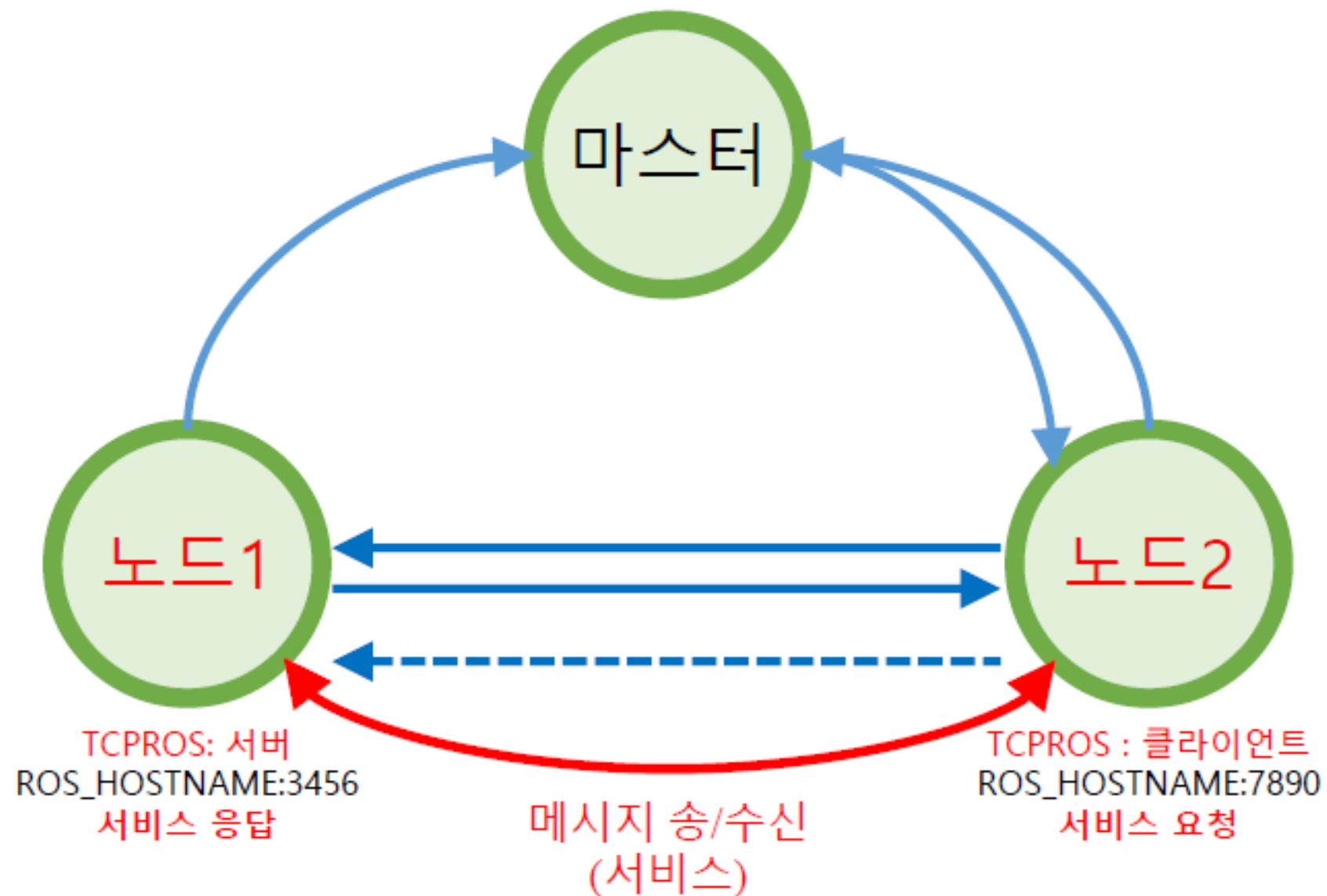
메시지 전송

➔ Publisher 노드는 subscriber 노드에게 메시지 전송 (topic)

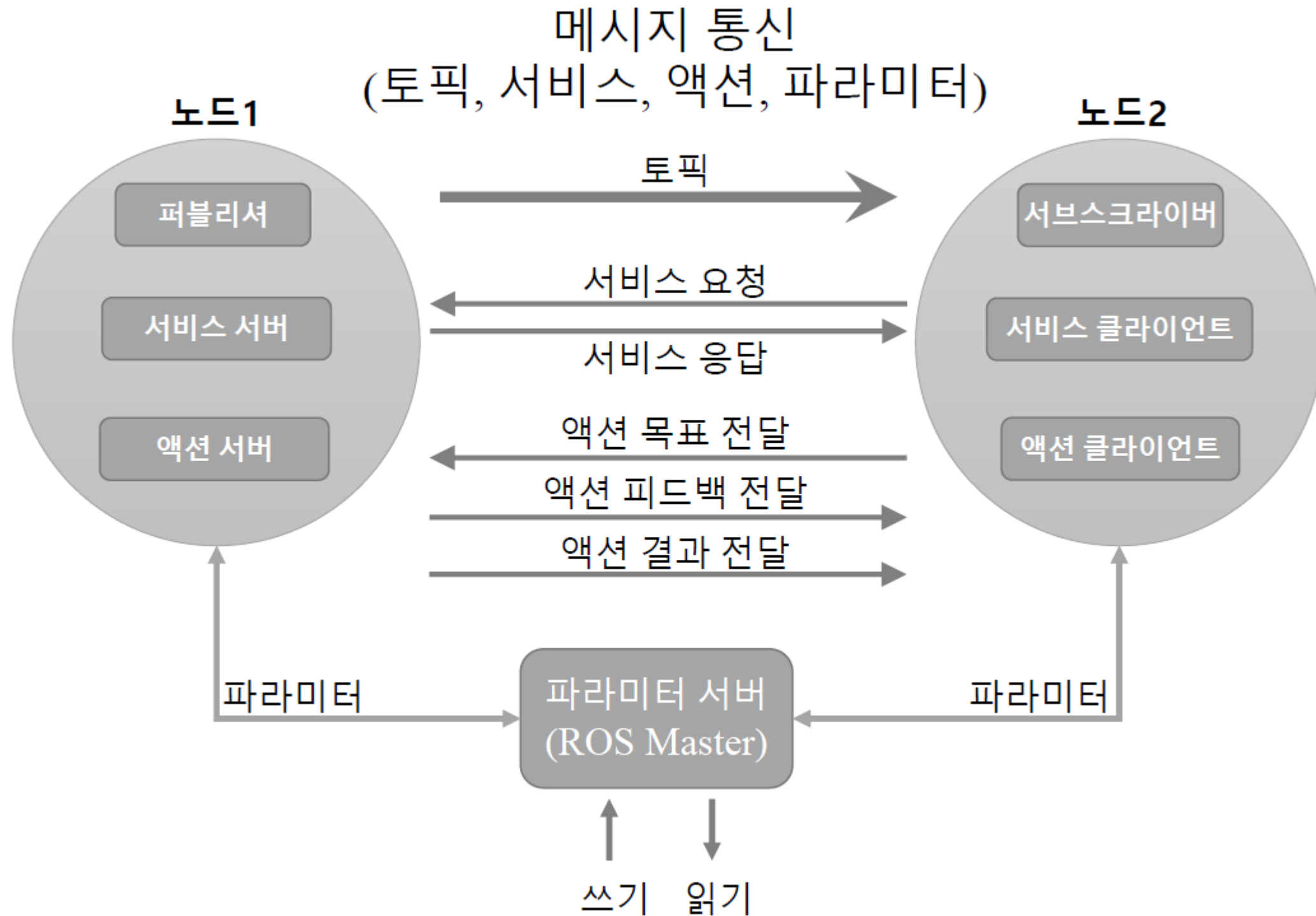


Service 요청 및 응답

➔ 1회에 한해 접속, 서비스 요청/응답 수행되고 서로간 접속 해제



ROS 개념



ROS 명령어 - 셸

명령어	중요도	명령어 풀이	세부 설명
roscd	★★★	ros+cd(changes directory)	지정한 ROS 패키지의 디렉터리로 이동
rosls	★☆☆	ros+ls(lists files)	ROS 패키지의 파일 목록 확인
rosed	★☆☆	ros+ed(editor)	ROS 패키지의 파일 편집
roscp	★☆☆	ros+cp(copies files)	ROS 패키지의 파일 복사
rospd	☆☆☆	ros+pushd	ROS 디렉터리 인덱스에 디렉터리 추가
rosd	☆☆☆	ros+directory	ROS 디렉터리 인덱스 확인

ROS 명령어 - 실행

명령어	중요도	명령어 풀이	세부 설명
roscore	★★★★	ros+core	<ul style="list-style-type: none">- master(ROS 네임 서비스)- rosout(로그 기록)- parameter server(파라미터 관리)
roslaunch	★★★★	ros+launch	노드를 여러 개 실행 및 실행 옵션 설정
rosclean	★★★☆☆	ros+clean	ROS 로그 파일을 검사하거나 삭제

ROS 명령어 - 정보

명령어	중요도	명령어 풀이	세부 설명
rostopic	★★★★	ros+topic	ROS 토픽 정보 확인
rosservice	★★★★	ros+service	ROS 서비스 정보 확인
roscall	★★★★	ros+node	ROS 노드 정보 확인
rosparam	★★★★	ros+param(parameter)	ROS 파라미터 정보 확인, 수정
rosbag	★★★★	ros+bag	ROS 메시지 기록, 재생
rosmmsg	★★★☆☆	ros+msg	ROS 메시지 정보 확인
rossrv	★★★☆☆	ros+srv	ROS 서비스 정보 확인
rosverison	★★☆☆☆	ros+version	ROS 패키지 및 배포 릴리즈 버전 정보 확인
roswtf	☆☆☆☆	ros+wtf	ROS 시스템 검사

ROS 명령어 - catkin

명령어	중요도	세부 설명
catkin_create_pkg	★★★	패키지 자동 생성
catkin_make	★★★	캐킨 빌드 시스템에 기반을 둔 빌드
catkin_eclipse	★★☆	캐킨 빌드 시스템으로 생성한 패키지를 이클립스에서 사용할 수 있게 변경
catkin_prepare_release	★★☆	릴리즈할 때 사용되는 로그 정리 및 버전 태깅
catkin_generate_changelog	★★☆	릴리즈할 때 CHANGELOG.rst 파일 생성 또는 업데이트
catkin_init_workspace	★★☆	캐킨 빌드 시스템의 작업 폴더 초기화
catkin_find	★☆☆	캐킨 검색

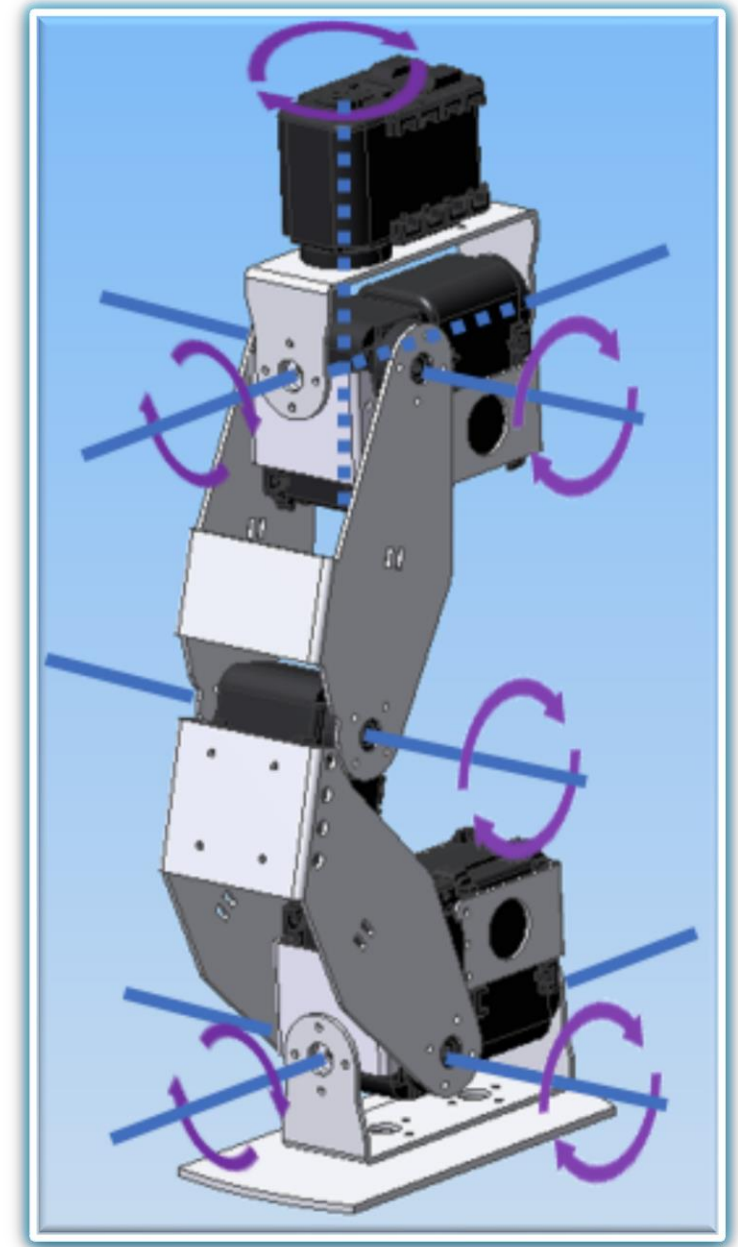
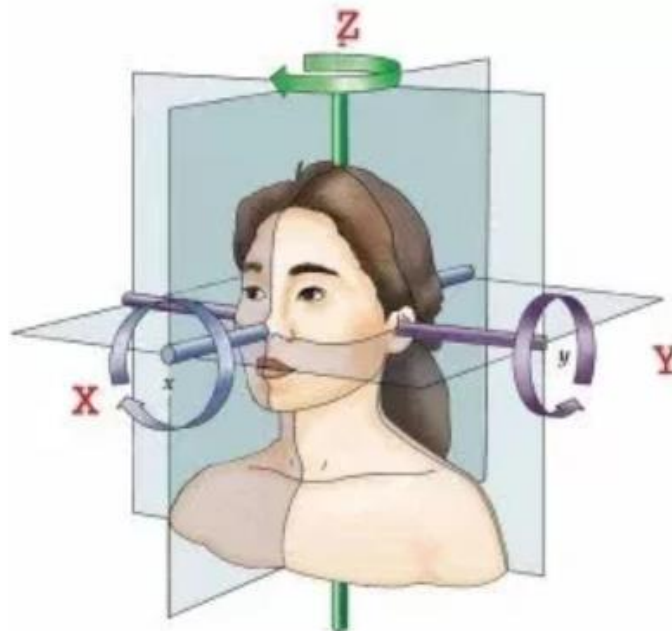
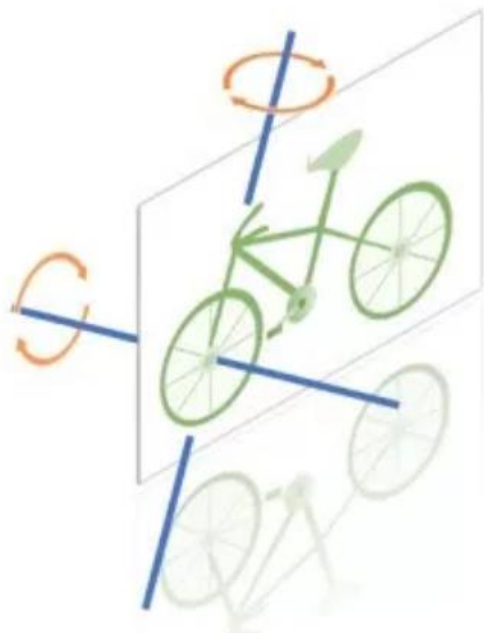
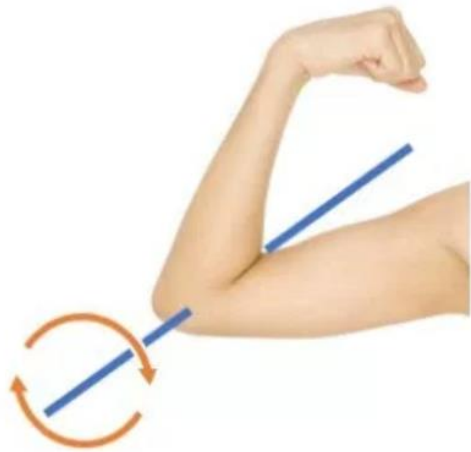
ROS 명령어 - 패키지

명령어	중요도	명령어 풀이	세부 설명
rospack	★★★	ros+pack(age)	ROS 패키지와 관련된 정보 보기
roinstall	★★☆	ros+install	ROS 추가 패키지 설치
rosdep	★★☆	ros+dep(endencies)	해당 패키지의 의존성 파일 설치
rosllocate	☆☆☆	ros+locate	ROS 패키지 정보 관련 명령어
roscrcreate-pkg	☆☆☆	ros+create-pkg	ROS 패키지 자동 생성(구 rosbuilt 시스템에서 사용)
rosmake	☆☆☆	ros+make	ROS 패키지를 빌드(구 rosbuilt 시스템에서 사용)

TF(transform)

- 좌표 변환

1. 하나의 DOF에 하나의 서보모터 사용
2. x,y,z 축에 대한 회전



<로봇 다리에 사용되는 모터>

**많이 해보아야
실력이 는다!**