

# ROS

# Robot Operating System

- Installation & Setup -

Department of Electrical Engineering, Incheon National University  
**Cheolmin Jeong**

2024. 8. 12.

# 강사 소개



## ■ 정철민 (Cheolmin Jeong)

- ~2023: 인천대학교 전기공학과 공학석사
- 2023~현재: 인천대학교 전기공학과 박사과정 재학 중
- 무인지능 시스템제어 연구실 소속  
(지도교수: 강창묵 교수, <https://uniconlab.wixsite.com/main>)



- 관심분야: 최적제어, 선형제어, 예측제어, 고장진단, 자율주행, SLAM
- 적용플랫폼: 자율주행차량, 모바일로봇, 드론, 4족보행로봇 등
- Projects:

- 자율주행 전동휠체어 핵심 기술 개발
- 순찰로봇의 도심지 적용을 위한 실외 자율주행 기술 개발
- 인천공항 폐기물 이송 로봇 개발
- UAV 기체 상태 이상 진단 검증 및 연계 모듈 개발
- 다목적 개인이동형 서비스 로봇 플랫폼 기술개발 및 실증
- 자율주행 전동휠체어 의료기기인증 및 표준안을 위한 기능 구현
- ...



# Contents

## 1. Linux

- 1) Linux 개요
- 2) Ubuntu
- 3) Ubuntu 이미지 파일 및 부팅 USB 세팅
- 4) 디스크 파티션 분할
- 5) Ubuntu 18.04 LTS 설치
- 6) 사용 언어 및 한영 전환기 설정
- 7) 주요 명령어

## 2. ROS

- 1) ROS 개요
- 2) ROS Melodic 설치
- 3) ROS1 기본 용어
- 4) ROS1 기본 명령어
- 5) 개발 환경 구축

## Appendix

# Linux



# 1. Linux

## 1) Linux 개요

- 정의

- 1991년 당시 핀란드 헬싱키 공대생이었던 리누스 토발즈(Linus Torvalds)에 의해 오픈소스로 개발된 개인용 컴퓨터, 서버, 메인 프레임, 모바일 기기, 임베디드 기기를 위한 Unix 호환 운영체제
- 리눅스가 리누스 토발즈에 의해 처음 개발된 당시에는 지금과 같은 파일 관리, 네트워크 관리, X Window, 개발자 도구 등의 다양한 패키지를 포함하는 운영체제의 형태는 아니었고, 단지 리눅스 커널만을 의미했음
- 현재는 FSF(Free Software Foundation)가 GNU 프로젝트를 통해 리눅스 커널 기반으로 다양한 라이브러리와 애플리케이션이 포함된 리눅스 배포판을 제공하고 있으며, 이를 강조하고자 특별히 GNU/Linux라고도 부름



# 1. Linux

## 1) Linux 개요

### ● 장점

- 오픈소스이기 때문에 상용 운영체제 대비 라이선스 비용이 들지 않아 경제적
- 오픈소스 정신을 지지하는 전세계의 수많은 개발자들이 공개된 리눅스의 커널 소스를 수정하고 개량함으로써 운영체제를 더욱 안정적으로 개선하고 보안성을 강화하고 있음
- 리눅스는 이더넷(Ethernet), SLIP, PPP 등의 다양한 네트워크 환경을 지원하며 TCP/IP, IPX 등의 네트워크 프로토콜도 지원
- 리눅스는 서버, 개발용, PC용 등 다양한 목적으로 사용할 수 있고, 이에 따른 다양한 배포판이 존재
- 임베디드 리눅스와 같이 특정 하드웨어에 맞게 커널의 기능을 더하거나 덜어내어 제품에 맞게 커스터마이징도 가능

### ● 단점

- 전세계의 개발자들이 리눅스를 발전시키고 있지만 리눅스를 사용함에 있어 발생하는 여러 사용자들의 이슈를 해결해 줄 수 있는 체계적인 기술지원 네트워크가 없음
- 다양한 오픈소스 소프트웨어가 존재하지만 마이크로소프트 오피스와 같은 쓸만한 리눅스용 상용 소프트웨어가 부족함
- 최신 하드웨어 기기에 대한 디바이스 드라이버 지원이 느림. 하드웨어 제조사에서 리눅스용 드라이버를 제공하는 경우에는 문제가 없지만 그렇지 않은 경우에는 오픈소스 개발자들이 하드웨어 기기를 역분석(reverse engineering)하여 드라이버를 만들어내는 데까지 오랜 시간이 걸림

# 1. Linux

## 1) Linux 개요

- 활용 분야

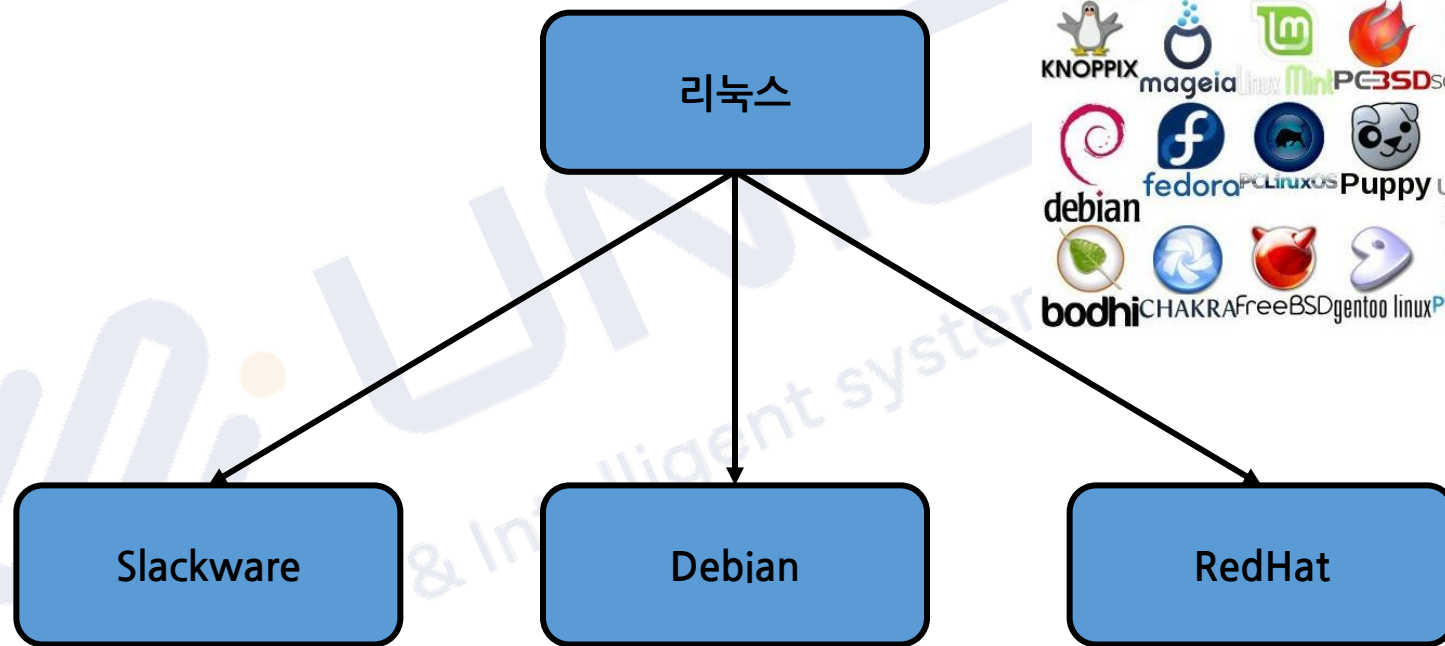
- 서버
- 메인 프레임
- 스마트 디바이스 (스마트폰, 태블릿, 스마트 TV, IVI(In-Vehicle Infotainment) 시스템 등)
- 임베디드 디바이스 (임베디드 보드, 각종 센서류, IoT 등)
- 게이밍 디바이스
- ...

퍼블릭 클라우드 컴퓨팅 워크로드의 90%, 스마트폰의 82%, 임베디드 기기의 62%, 슈퍼 컴퓨터 시장의 99%가 Linux로 작동

# 1. Linux

## 1) Linux 개요

- 리눅스 배포판



- 가장 오래된 배포판
- 가장 유닉스다운 리눅스 배포판

- 기업, 재단보다는 자발적 커뮤니티에 의한 배포판
- APT 패키지 관리자를 통한 패키지 설치 및 업그레이드의 편리함

- 고객 유료 서비스를 통한 수익 창출을 추구
- 일반 사용자를 위한 무료 배포판 제공





# 1. Linux

## 1) Linux 개요

### ● 리눅스 배포판

분류	종류	설명
RedHat 계열	RHEL (RedHat Enterprise Linux)	<ul style="list-style-type: none"> <li>페도라를 기반으로 레드햇에서 개발한 리눅스 배포판이며 안정성을 목적으로 제작되었고 유료 라이선스로 제공되며 레드햇이 기술지원을 제공</li> </ul>
	Fedora	<ul style="list-style-type: none"> <li>레드햇의 후원과 페도라 커뮤니티의 지원 아래 범용 목적으로 개발된 것으로, RPM 기반의 소프트웨어가 결합된 리눅스 배포판</li> <li>MeeGo, SailFish OS, Tizen</li> </ul>
	CentOS (Community ENTERprise Operating System)	<ul style="list-style-type: none"> <li>페도라가 최신기술을 담아내는데 집중한 나머지 RHEL을 제대로 반영하지 못하자 RHEL을 완벽하게 호환하여 무료 기업용 컴퓨팅 플랫폼 제공 목적으로 개발된 리눅스 배포판</li> </ul>
	Oracle Linux	<ul style="list-style-type: none"> <li>2006년부터 GNU GPL 라이선스에 따라 레드햇 소스코드를 공유하면서 오라클 어플라이언스에 최적화된 리눅스 배포판</li> </ul>
	Scientific Linux	<ul style="list-style-type: none"> <li>RHEL 기반의 자유 및 오픈소스 운영 체제로서 상용 기업형 배포판에 근접하도록 만들기 위한 목적으로 페르미 국립 가속기 연구소가 개발한 리눅스 배포판</li> </ul>
Debian 계열	Ubuntu	<ul style="list-style-type: none"> <li>마크 셔틀워스가 우분투 보급을 위해 세운 영국의 컴퓨터 소프트웨어 회사인 캐노니컬 유한회사(Canonical Ltd)에서 개발한 리눅스 배포판</li> <li>전세계 사람 누구나 리눅스를 쉽게 사용하자는 미션을 바탕으로 사용자 편의성에 초점을 맞춘 리눅스</li> </ul>
	Raspbian	<ul style="list-style-type: none"> <li>마이크 톰슨과 피터그린이 독립 프로젝트로 시작한 라즈비안은 2012년 6월 처음 완성된 이래로 2015년부터 라즈베리파이 재단에 의해 공식 지원을 받고 있는 라즈베리파이 전용 리눅스</li> </ul>
	ChromeOS / Chromium OS	<ul style="list-style-type: none"> <li>크롬OS는 구글이 설계하고 개발한 웹 애플리케이션 전용 운영체제로 사용자 인터페이스는 크롬 브라우저와 유사</li> <li>사용자가 대부분의 시간을 브라우저를 이용하여 인터넷을 탐색한다는 점에 착안을 두어 개발한 리눅스 기반 운영체제</li> </ul>
Slackware 계열	Slackware	<ul style="list-style-type: none"> <li>가장 먼저 대중화된 현존하는 가장 오래된 배포판으로 1993년 패트릭 볼커딩에 의해 개발</li> <li>간결함을 설계 철학으로 하여 유닉스 자체 학습에 적합</li> </ul>
	openSUSE	<ul style="list-style-type: none"> <li>슬랙웨어 리눅스로부터 파생되었으며 독일에서 제작하였고 유럽에서 많이 사용</li> <li>유료 라이선스인 SUSE Linux Enterprise와 무료 버전인 openSUSE 두 가지로 배포</li> </ul>
	Vector Linux	<ul style="list-style-type: none"> <li>로버트 S. 란게와 다렐 스타뎀에 의해 개발된 SOHO, Standard, Light, Live판을 갖는 슬랙웨어 계열 리눅스 운영체제</li> </ul>

# 1. Linux

## 1) Linux 개요

### ● 리눅스 배포판

분류	종류	설명
Android 계열	Android	<ul style="list-style-type: none"> <li>휴대용 기기에 최적화된 리눅스 기반 모바일 운영체제</li> <li>리눅스 커널을 기반으로 안드로이드 런타임 실행</li> <li>안드로이드 런타임상에는 웹브라우저, 이메일 클라이언트, 단문 메시지 서비스(SMS) 등 다양한 응용 프로그램이 구동하고 이러한 앱을 위한 풍부한 런타임 라이브러리 제공</li> </ul>
	AOSP (Android Open Source Project)	<ul style="list-style-type: none"> <li>안드로이드는 구글에 의해 개발되지만 소스코드는 AOSP를 통해 운영</li> <li>AOSP를 통해 다양한 디바이스에 최적화된 안드로이드를 포팅 및 커스터마이징할 수 있음</li> </ul>
국내 계열	하모니카	<ul style="list-style-type: none"> <li>2014년 과기정통부의 전신인 미래창조과학부에서 발표한 “공개 SW 활성화 계획”의 일환으로 정부가 주도로 개발한 리눅스 민트 기반 개방형 운영체제</li> <li>기존 배포된 리눅스 민트보다 한글화가 잘 되어 있음</li> </ul>
	넘버원 리눅스	<ul style="list-style-type: none"> <li>독자적인 토종 리눅스 개발을 목표로 PCLinux 기반 KDE 환경을 제공하는 리눅스 배포판</li> </ul>
	구름OS	<ul style="list-style-type: none"> <li>과학기술정보통신부와 한국전자통신연구원(ETRI) 산하 국가보안기술연구소 및 한글과 컴퓨터가 개발한 개방형 OS</li> </ul>

# 1. Linux

## 2) Ubuntu

- 편리한 설치와 사용법으로 초보자들이 연습하기 좋은 배포판
- 6개월 단위로 새 버전 발표, 2년 주기로 LTS(Long Term Support: 5년 간 기술지원 제공) 버전 발표

버전	코드네임	배포일	커널 버전	지원 종료일
6.06 LTS	Dapper Drake	2006년 06월 01일	2.6.15	2009년 07월
8.04 LTS	Hardy Heron	2008년 04월 24일	2.6.24	2011년 05월
10.04 LTS	Lucid Lynx	2010년 04월 29일	2.6.32	2013년 05월
12.04 LTS	Precise Pangolin	2012년 04월 26일	3.2.0	2017년 04월
14.04 LTS	Trusty Tahr	2014년 04월 17일	3.13.0	2019년 04월
16.04 LTS	Xenial Xerus	2016년 04월 21일	4.4.0	2021년 04월
<b>18.04 LTS</b>	<b>Bionic Beaver</b>	<b>2018년 04월 27일</b>	<b>4.15</b>	<b>2023년 04월</b>
20.04 LTS	Focal Fossa	2020년 04월 23일	5.4	2025년 04월 예정
22.04 LTS	Jammy Jellyfish	2022년 02월 22일	5.15	2027년 04월 예정
24.04 LTS	Noble Numbat	2024년 04월 25일	6.8	2029년 04월 예정

- 시스템 요구 사항

CPU	2 GHz dual core processor
메모리	4 GB
디스크 여유 공간	25 GB
그래픽카드 해상도	1024x768



# 1. Linux

## 3) Ubuntu 이미지 파일 및 부팅 USB 세팅

- Ubuntu 18.04 LTS 이미지 파일 다운로드

[Ubuntu 18.04 LTS image file link](#)



### Select an image

Ubuntu is distributed on three types of images described below.

#### Desktop image

The desktop image allows you to try Ubuntu without changing your computer at all, and at your option to install it permanently later. This type of image is what most people will want to use. You will need at least 1024MiB of RAM to install from this image.

#### 64-bit PC (AMD64) desktop image

Choose this if you have a computer based on the AMD64 or EM64T architecture (e.g., Athlon64, Opteron, EM64T Xeon, Core 2). Choose this if you are at all unsure.

#### Server install image

The server install image allows you to install Ubuntu permanently on a computer for use as a server. It will not install a graphical user interface.

#### 64-bit PC (AMD64) server install image

Choose this if you have a computer based on the AMD64 or EM64T architecture (e.g., Athlon64, Opteron, EM64T Xeon, Core 2). Choose this if you are at all unsure.

- Ubuntu 공식 홈페이지의 다운로드 페이지에서 **18.04 LTS 버전 desktop image**로 다운로드

# 1. Linux

## 3) Ubuntu 이미지 파일 및 부팅 USB 세팅

- Rufus 툴 다운로드

[Rufus link](#)



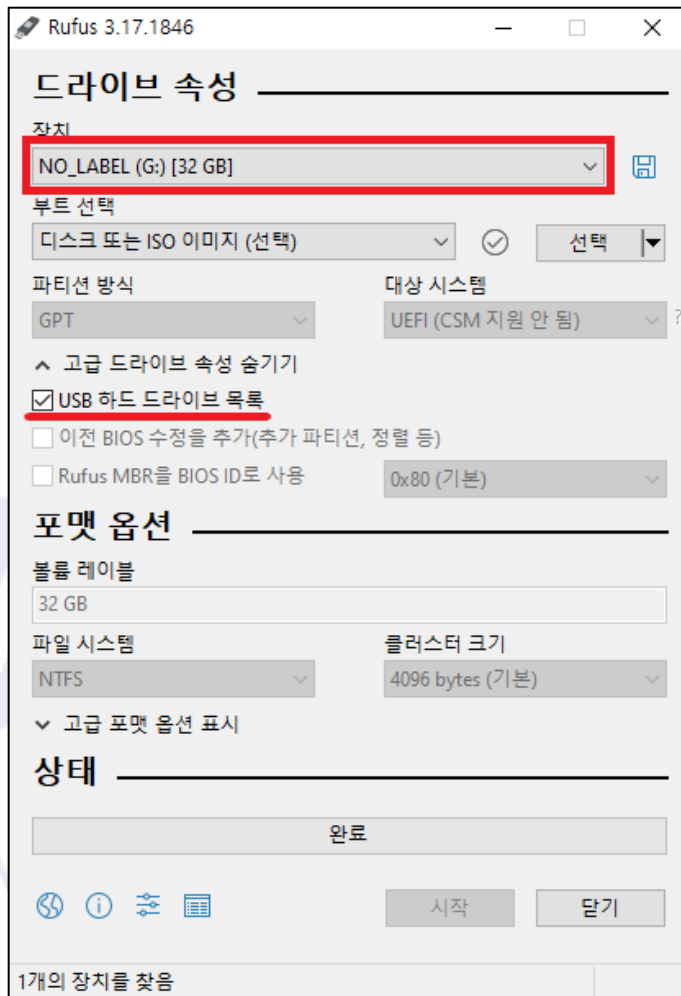
다운로드				
최신 릴리스:				
링크	유형	플랫폼	크기	날짜
<a href="#">rufus-4.5.exe</a>	표준	Windows x64	1.4 MB	2024.05.22
<a href="#">rufus-4.5p.exe</a>	포터블	Windows x64	1.4 MB	2024.05.22
<a href="#">rufus-4.5_x86.exe</a>	표준	Windows x86	1.5 MB	2024.05.22
<a href="#">rufus-4.5_arm64.exe</a>	표준	Windows ARM64	4.8 MB	2024.05.22

- 시스템 요구 사항 : Windows 8 이상
- Rufus 공식 홈페이지에서 최신 버전으로 다운로드
- Rufus는 부팅 가능한 iso 파일(Windows, Linux, UEFI 등)이 있을 때 간단하게 OS 설치용 USB를 만들 수 있는 오픈소스 툴
- 다른 부팅 USB 툴을 사용해도 무방

# 1. Linux

## 3) Ubuntu 이미지 파일 및 부팅 USB 세팅

- Ubuntu 설치 USB 세팅

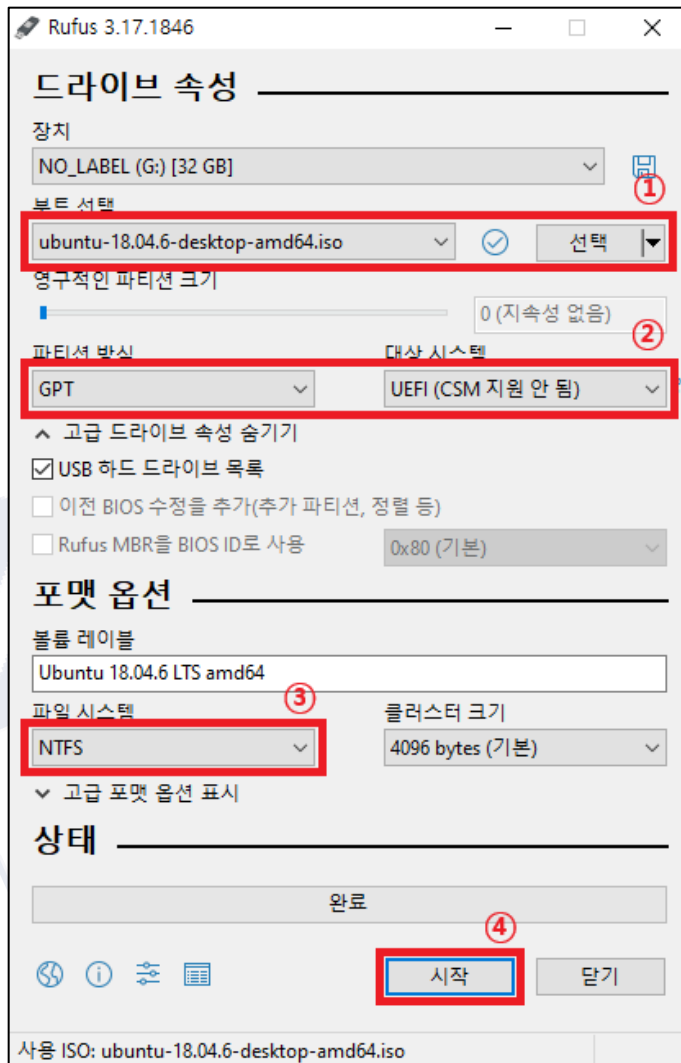


- 포맷가능한 USB 준비
- Ubuntu 설치용 USB를 PC에 장착한 후 Rufus 실행
- Rufus에서 USB 장치가 인식되었는지 확인
- 인식이 안 되었을 경우, 고급 드라이브 속성 숨기기에서 USB 하드 드라이브 목록을 체크한 후 다시 USB 장치 선택

# 1. Linux

## 3) Ubuntu 이미지 파일 및 부팅 USB 세팅

- Ubuntu 설치 USB 세팅



- 올바르게 USB 장치가 인식되었다면 다음과 같이 선택

① 부팅 선택에서 다운받은 Ubuntu iso 파일을 선택

② 파티션 구성은 GPT, 대상 시스템은 UEFI

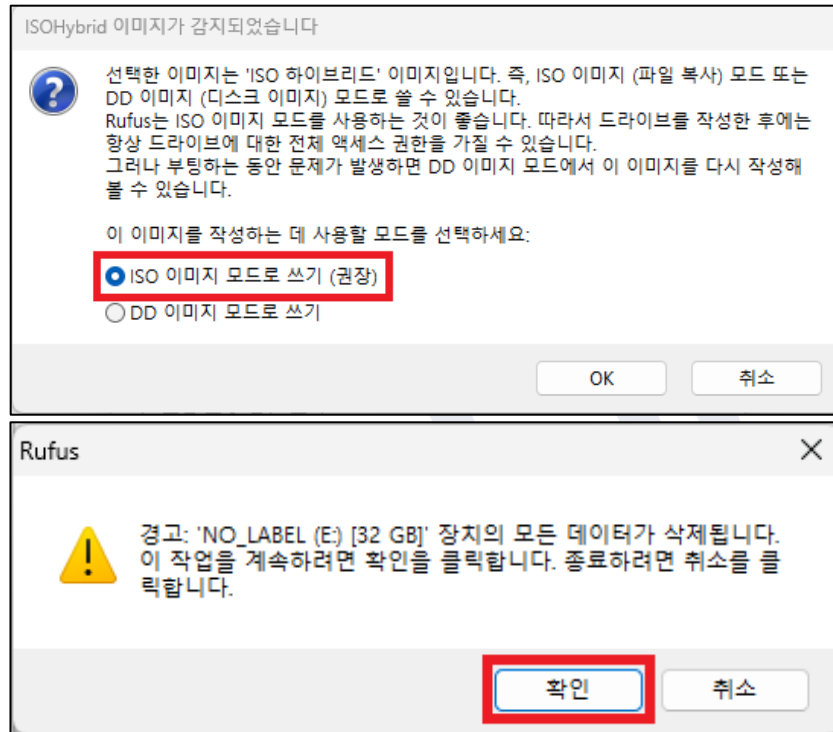
③ 파일 시스템은 NTFS

④ 선택이 완료되었으면 시작버튼을 눌러 USB장치 포맷과 Ubuntu 설치 USB 세팅을 시작

# 1. Linux

## 3) Ubuntu 이미지 파일 및 부팅 USB 세팅

- Ubuntu 설치 USB 세팅



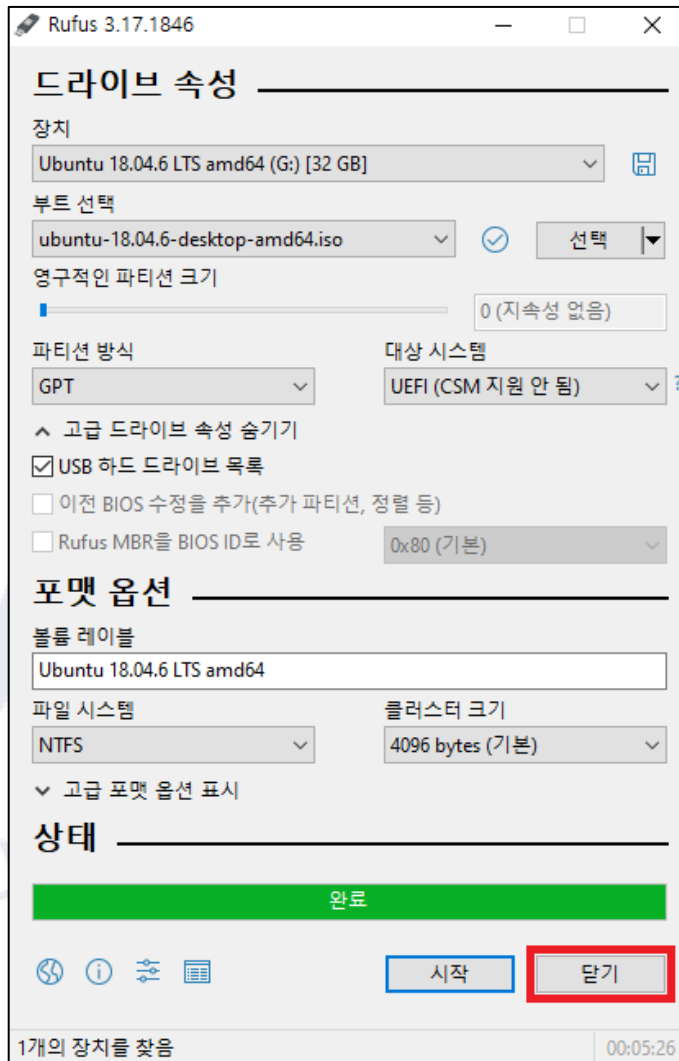
- PC의 제조사에 따라서 두 가지 또는 한 가지 경고문 발생
- 그림과 같이 선택하여 계속 진행



# 1. Linux

## 3) Ubuntu 이미지 파일 및 부팅 USB 세팅

- Ubuntu 설치 USB 세팅

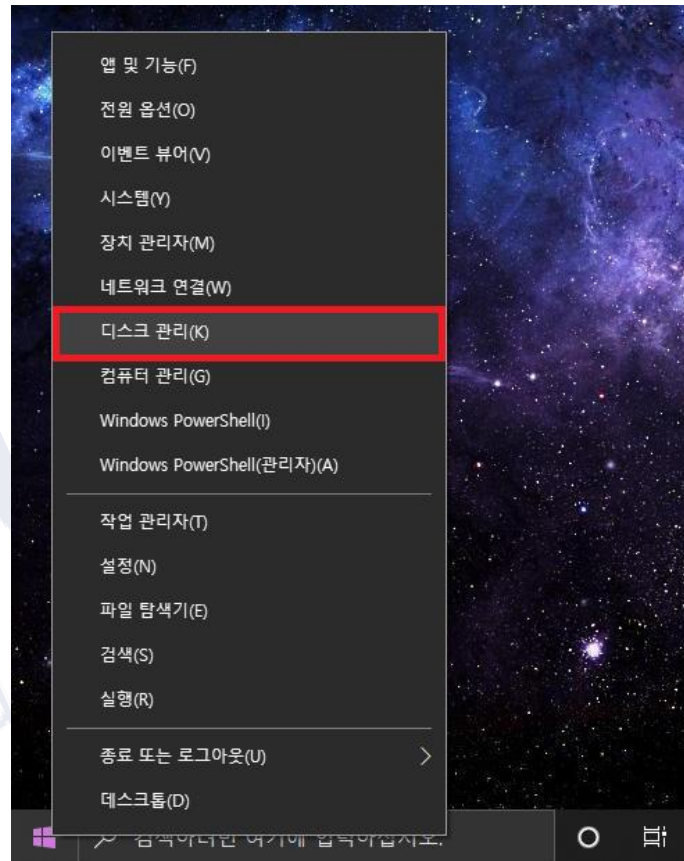


- USB 포맷과 iso 파일 복사가 끝나면 상태가 완료 또는 준비 상태로 변경
- 모두 완료된 후 닫기를 누르면 모든 과정이 끝나고 설치 USB 바로 사용 가능

# 1. Linux

## 4) 디스크 파티션 분할 (선택)

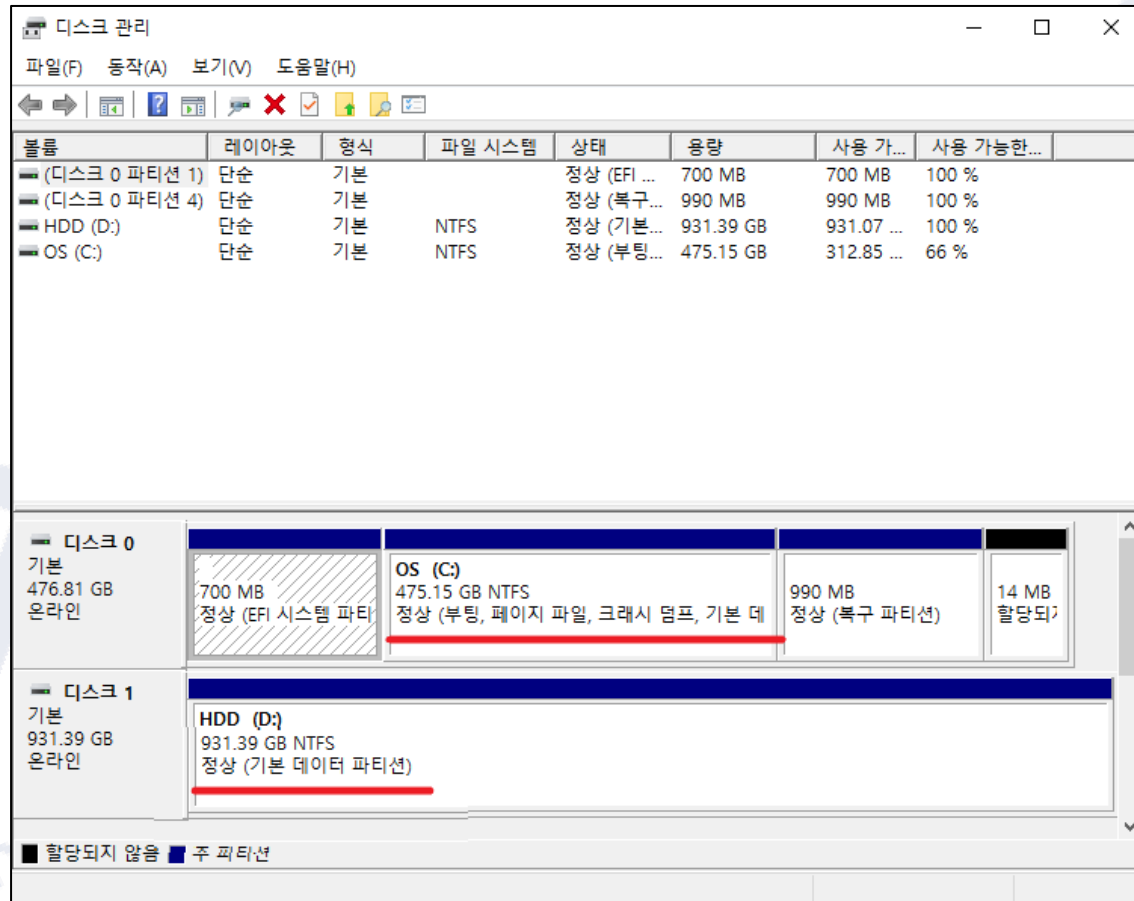
윈도우에서 멀티 운영체제로 ubuntu를 설치하는 경우에만 해당



- 시작 버튼에서 마우스 오른쪽버튼을 클릭한 후 디스크 관리를 선택

# 1. Linux

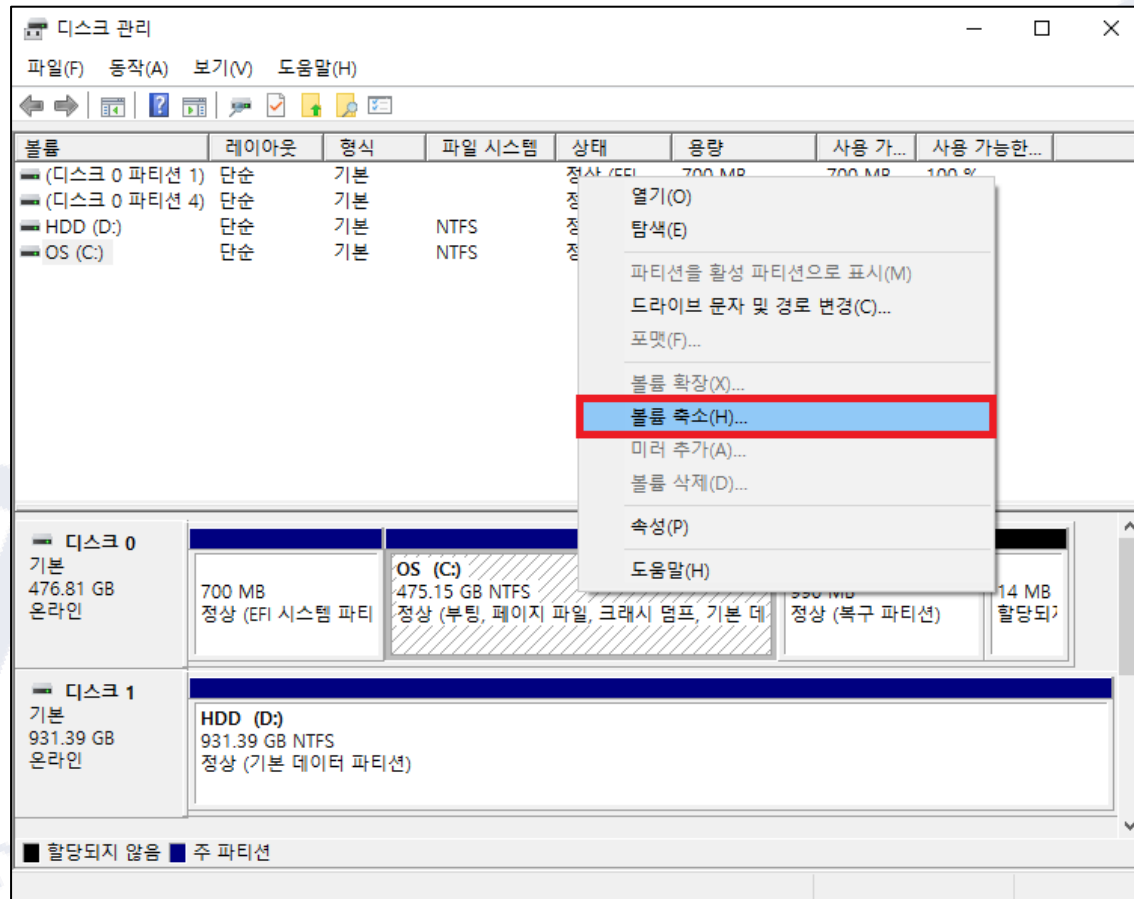
## 4) 디스크 파티션 분할 (선택)



- 디스크 관리창이 나타나면 자신의 PC의 드라이브 상태를 확인하고, 어떤 드라이브의 파티션을 분할하여 Ubuntu를 설치할 것인지 결정

# 1. Linux

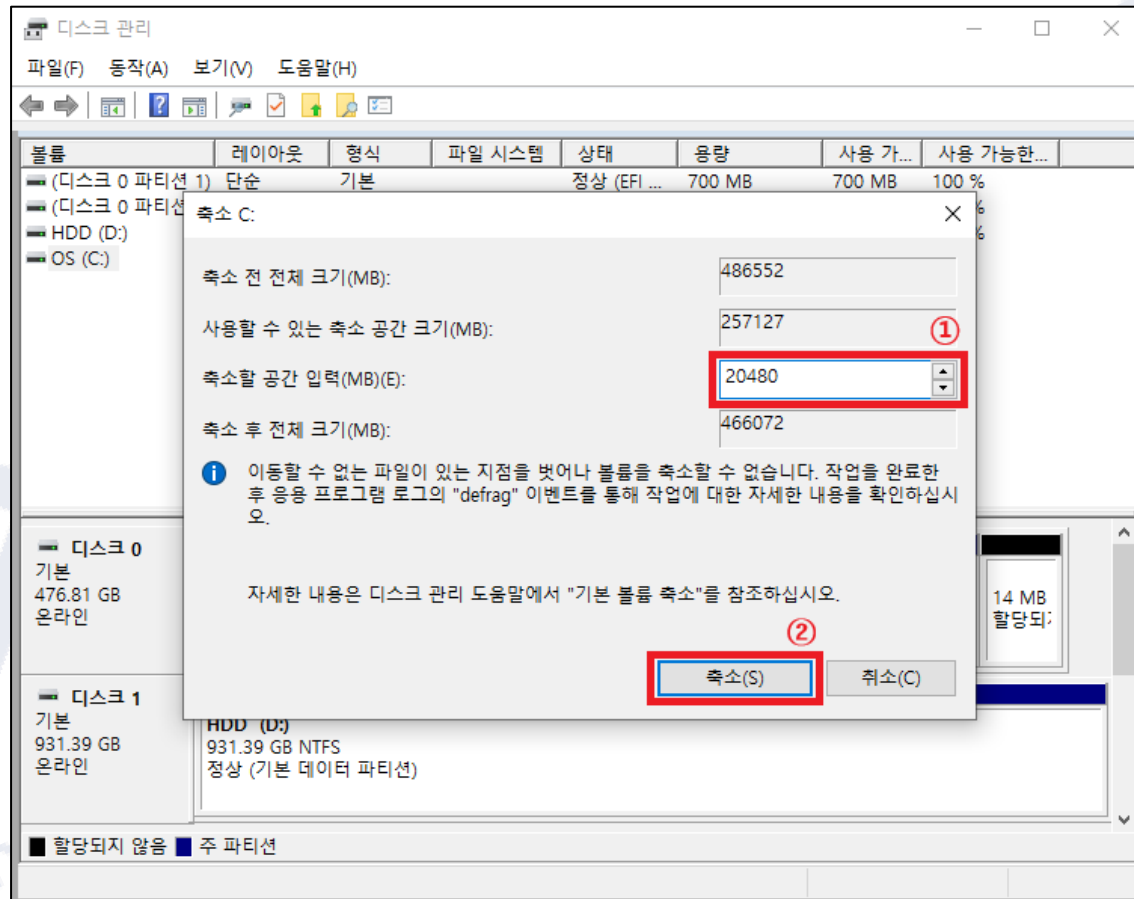
## 4) 디스크 파티션 분할 (선택)



- 파티션 분할을 진행할 드라이브의 주 파티션에 마우스 오른쪽버튼을 클릭
- 메뉴 중 **볼륨 축소**를 선택

# 1. Linux

## 4) 디스크 파티션 분할 (선택)



- 축소 설정 창이 나타나면 축소하고자 하는 디스크의 크기를 입력 후 **축소** 버튼을 클릭

# 1. Linux

## 4) 디스크 파티션 분할 (선택)

### ※ 디스크 할당 크기 결정

Ubuntu에서 사용할 디스크의 크기를 계산

ex) 20GB를 할당하고 싶을 때 :  $1024 \text{ (MB)} \times 20 = 20480 \text{ (MB)}$

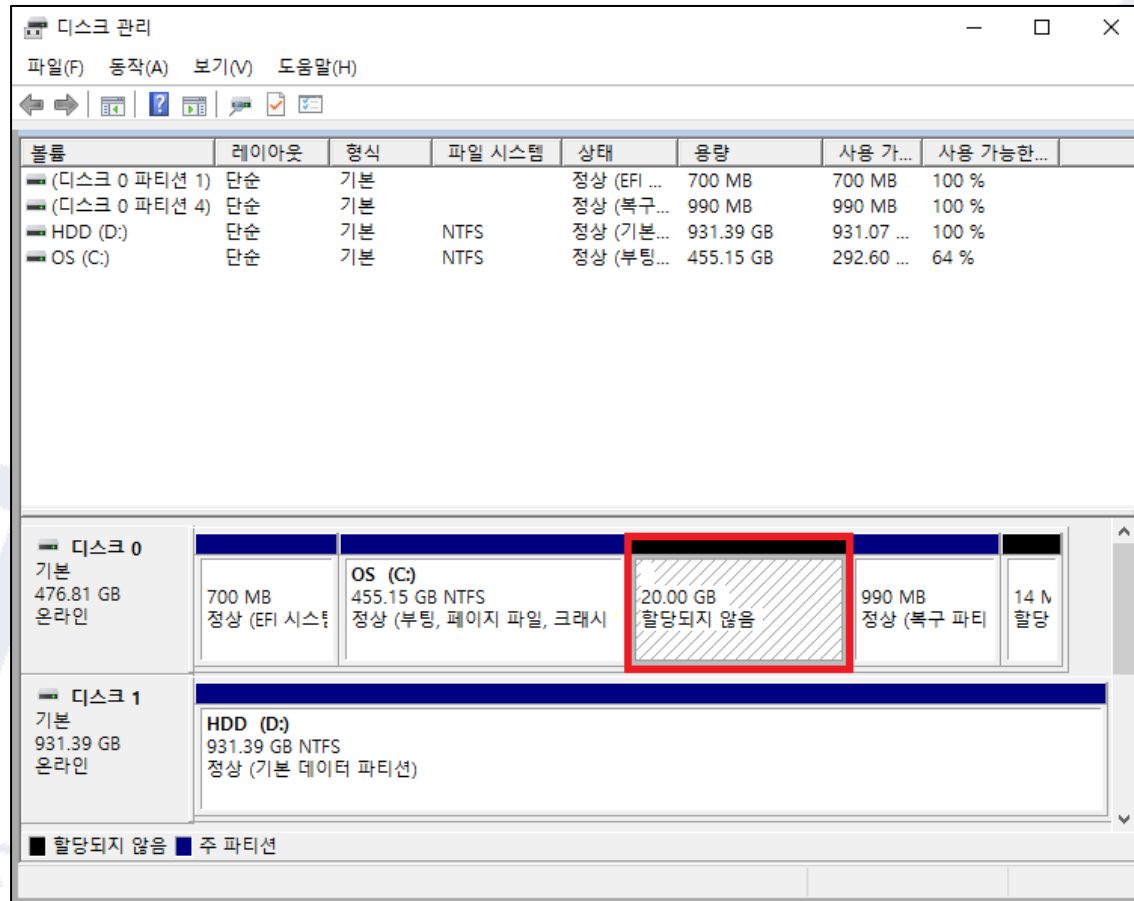
본인의 프로젝트 및 하드웨어 상황에 따라서 유동적으로 결정

### ※ 디스크 축소가 되지 않을 경우

축소 버튼을 눌렀을 때, 드라이브의 남은 공간이 충분한데도 불구하고 드라이브의 공간이 부족하다며 축소가 되지 않는 경우에는 드라이브에 이미 존재하는 시스템파일 등으로 인해 축소가 되지 않으므로 드라이브를 포맷한 후 사용하거나 새로운 드라이브에 재시도하여야 함

# 1. Linux

## 4) 디스크 파티션 분할 (선택)



- 볼륨 축소가 완료되면 디스크의 파티션이 분할된 것을 확인할 수 있음

# 1. Linux

## 5) Ubuntu 18.04 LTS 설치

### ※ 메인보드 bios 세팅 환경 진입 및 부팅 디스크 우선순위 변경

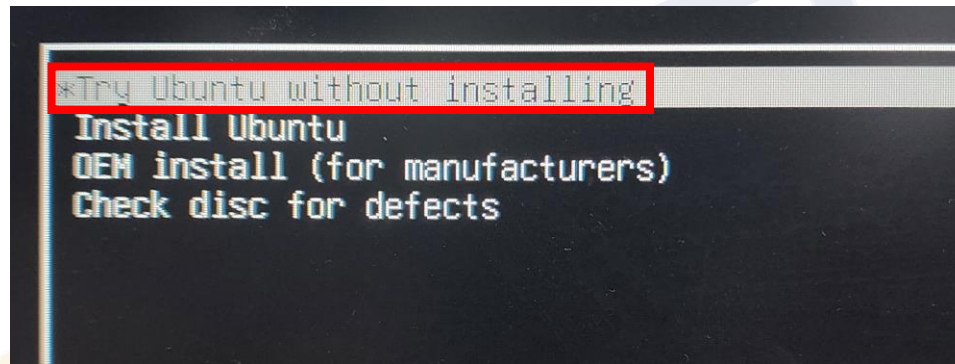
(수동으로 부팅 디스크의 우선순위를 변경할 때, 각 메인보드마다 진입 방식이나 세팅 환경이 다를 수 있으니 메인보드의 매뉴얼을 확인하고 진행하는 것을 권장)

- 세팅 완료된 Ubuntu 설치 USB를 PC에 장착된 상태로 재부팅
- PC의 부팅이 시작될 때, F2, F10, F12, del 키 중 메인보드 환경에 맞는 키를 입력하여 bios 세팅으로 진입
- Bios 세팅 환경에 진입 후, boot option 탭에서 Ubuntu 설치 USB의 부팅 디스크의 우선순위를 가장 높게 설정
- Save, exit 등을 통해 설정을 저장한 후 PC 재부팅
- Ubuntu 설치 시작



# 1. Linux

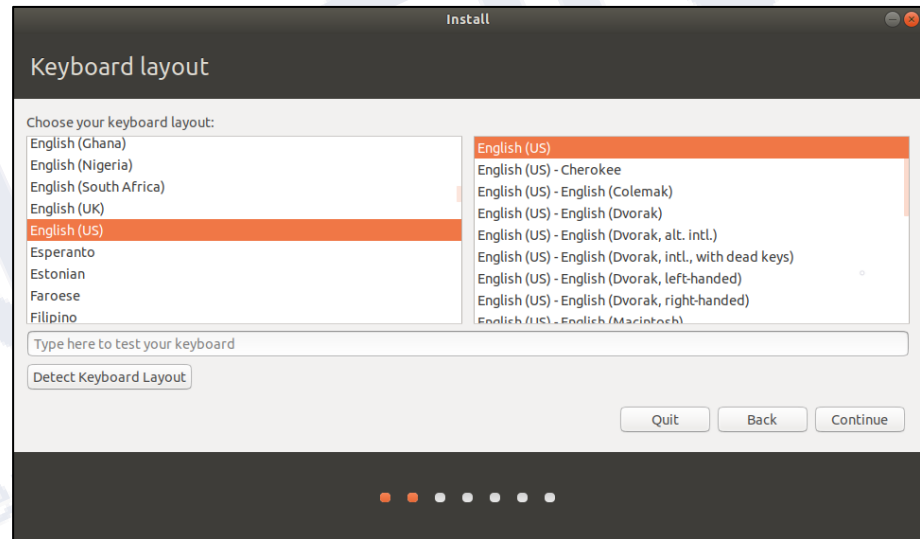
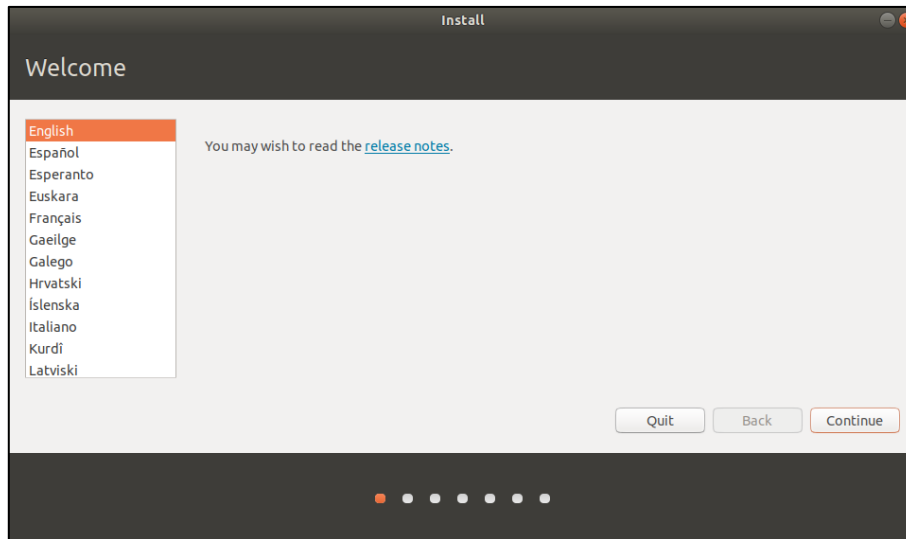
## 5) Ubuntu 18.04 LTS 설치



- 다음 두 가지 선택지 중 하나를 선택하여 설치 시작
  - Try Ubuntu: Ubuntu 환경에 먼저 진입한 다음 그 안에서 설치
  - Install Ubuntu: Ubuntu 설치를 바로 실행하는 항목

# 1. Linux

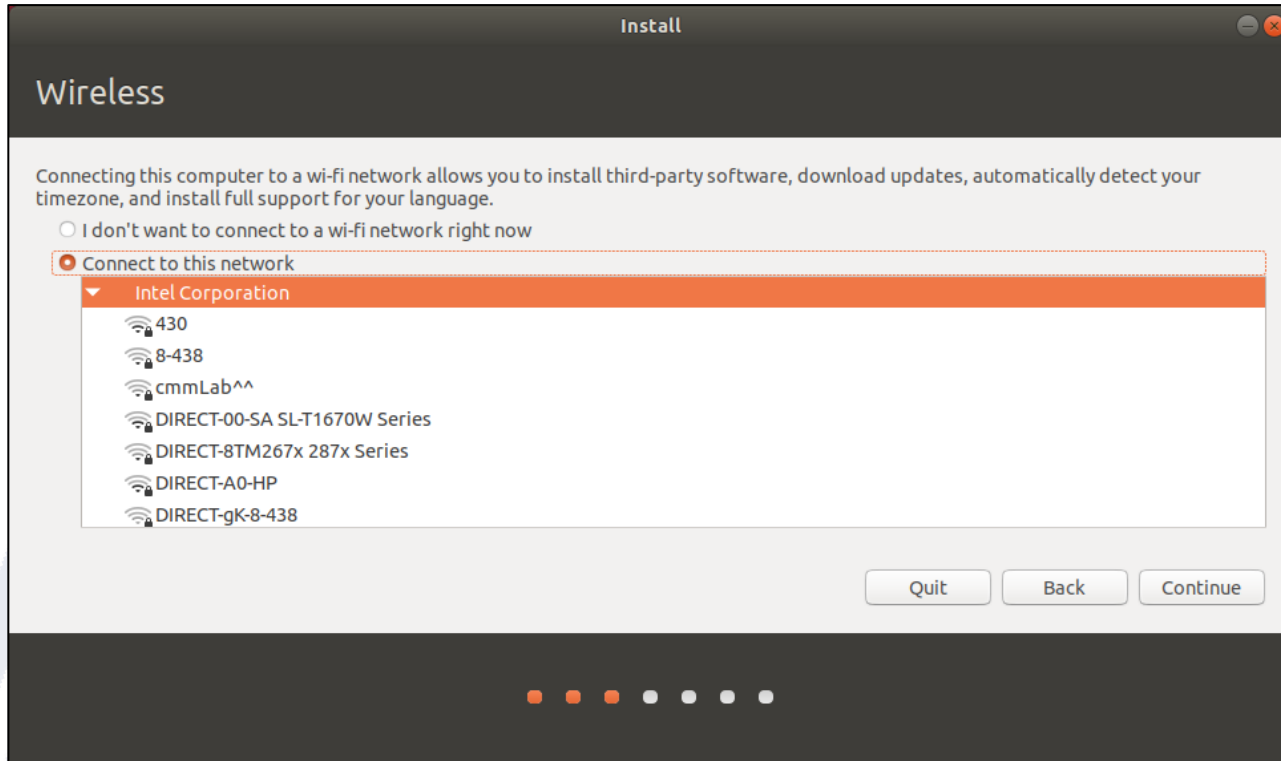
## 5) Ubuntu 18.04 LTS 설치



- 원하는 언어와 keyboard layout을 선택 (일반적으로 English)

# 1. Linux

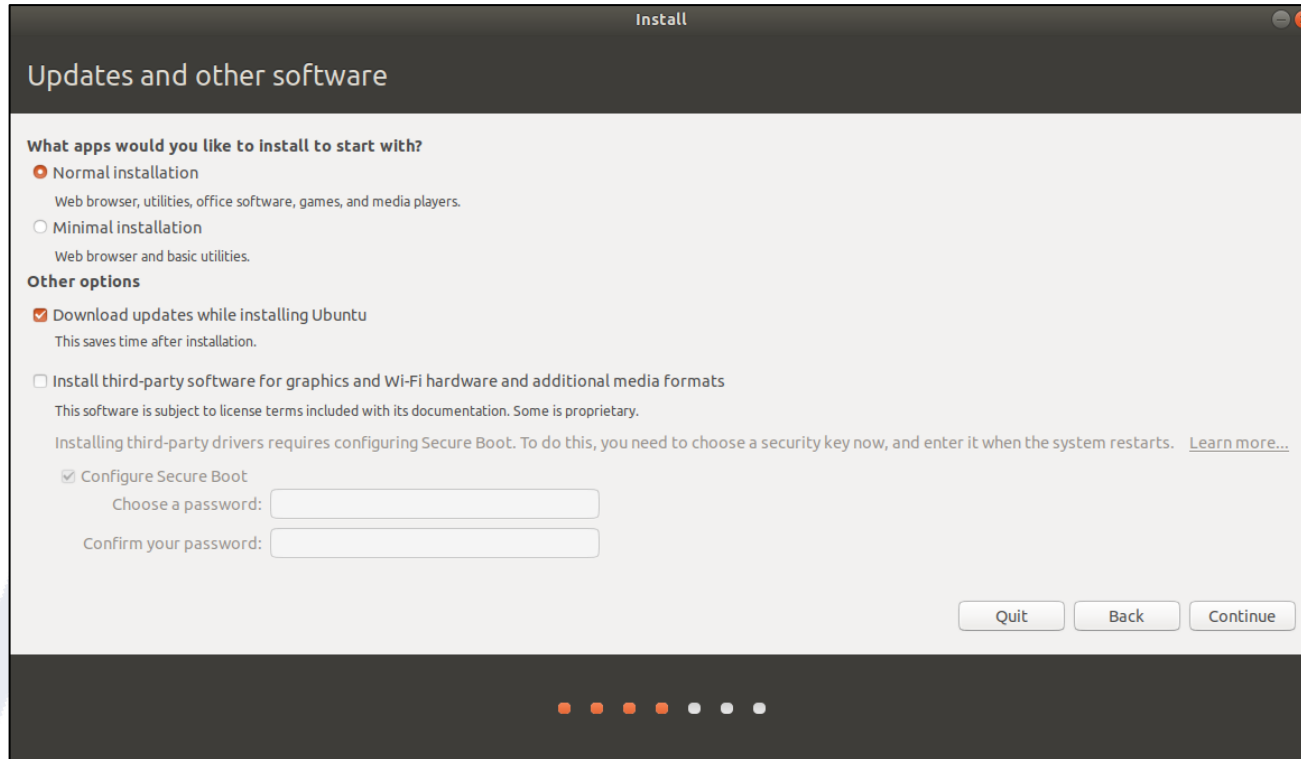
## 5) Ubuntu 18.04 LTS 설치



- 네트워크 연결을 원하는 경우 연결 (설치와 업데이트를 동시에 하기 위함)

# 1. Linux

## 5) Ubuntu 18.04 LTS 설치



- Normal installation과 Download updates while installing Ubuntu를 선택

What apps would you like to install to start with?

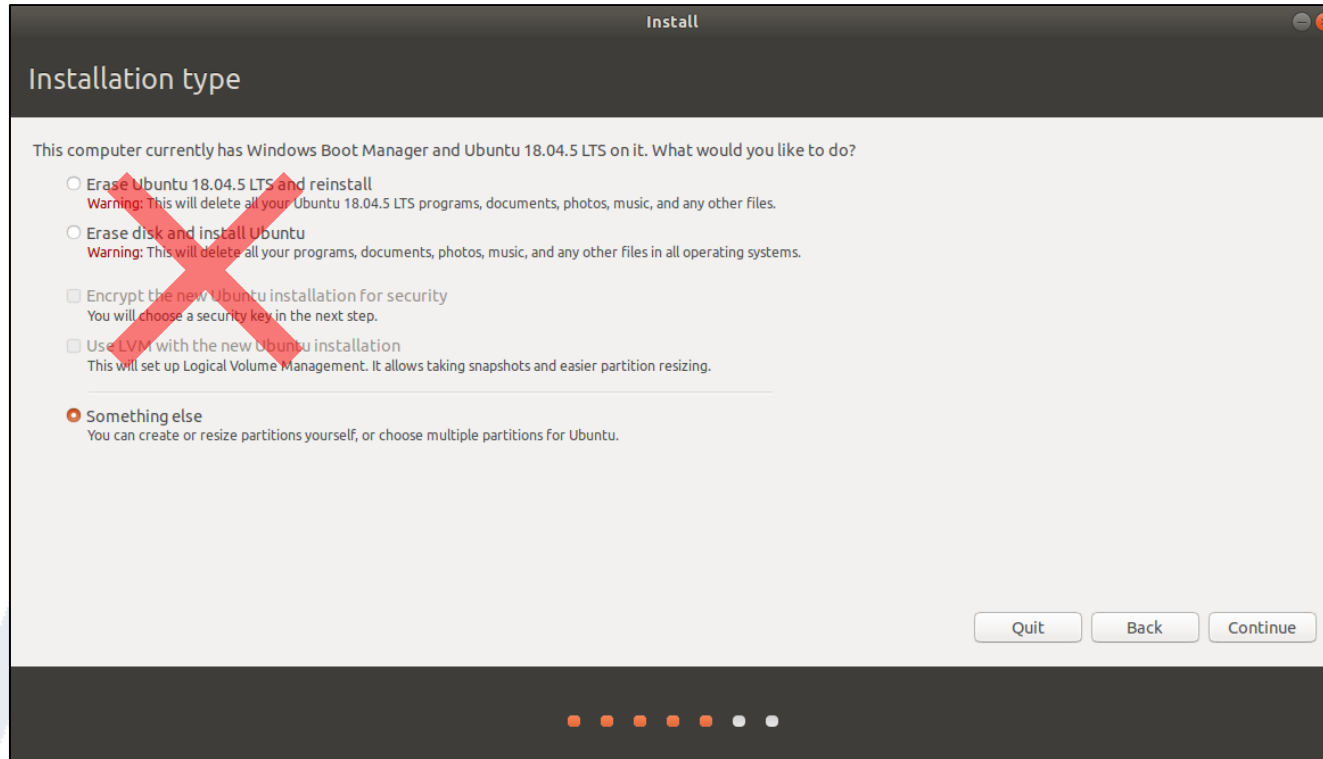
- Normal installation: 일반 설치
- Minimal installation: 최소 설치

Other options

- Download updates while installing Ubuntu: 네트워크가 연결되어 있는 경우, 우분투를 설치하는 동안 업데이트 다운로드
- Install third-party software for graphics and Wi-Fi hardware and additional media formats: 그래픽 및 Wi-Fi 하드웨어, MP3 및 추가 미디어 형식을 위한 타사 소프트웨어 설치 시스템에 인터넷이 연결되어 있는 경우, 이 옵션 선택

# 1. Linux

## 5) Ubuntu 18.04 LTS 설치



※ 설치 중 가장 주의가 필요한 단계 (두번째 선택지로 진행할 경우 기존 드라이브의 모든 데이터가 포맷됨)

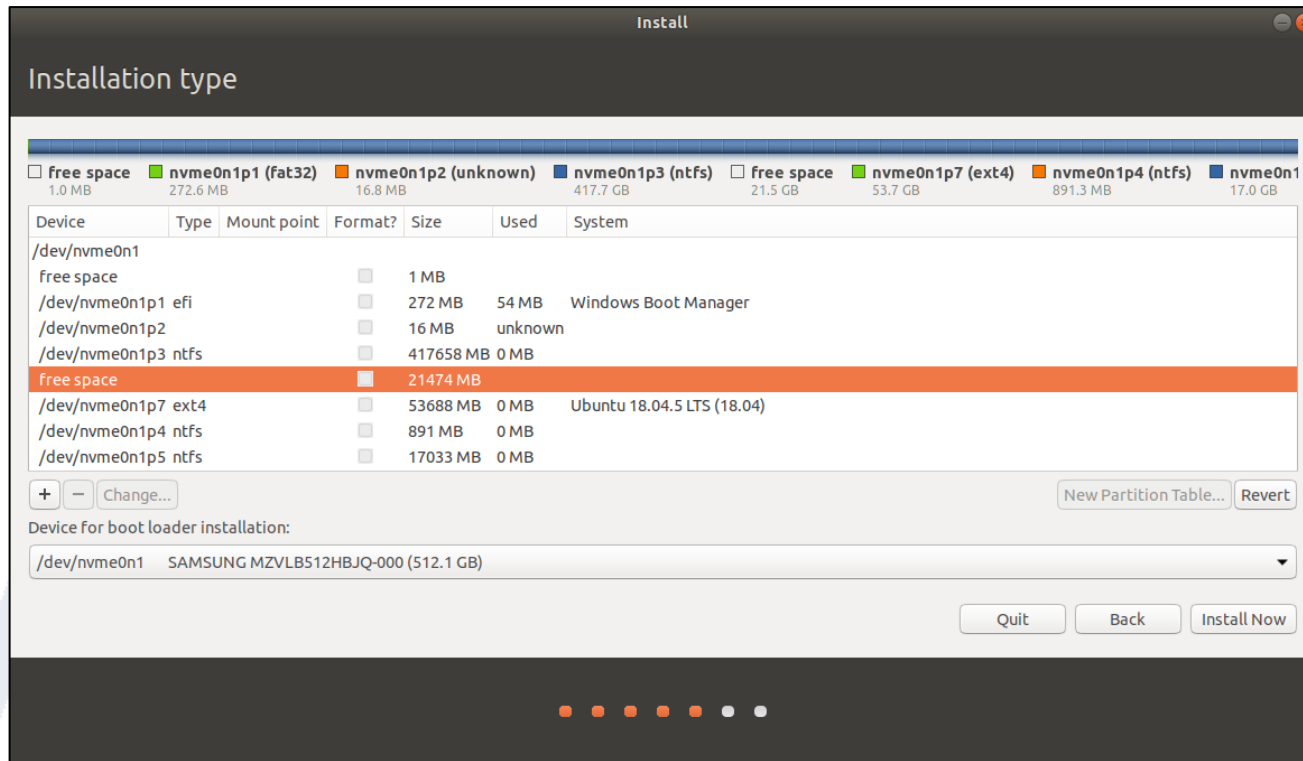
- 미리 분할해 놓은 파티션에 설치를 진행할 경우 **Something else**, 기존 드라이브의 모든 데이터를 지우고 Ubuntu를 설치하는 경우 **Erase disk and install Ubuntu**(파티션 선택 없이 바로 설치 시작) 선택

What would you like to do?

- Erase Ubuntu 18.04.5 LTS and reinstall: PC에 이전 버전의 Ubuntu가 이미 설치되어 있다면 나타나는 항목. 이전 버전의 Ubuntu를 삭제하고 새로 설치하려면 선택
- Erase disk and install Ubuntu: 드라이브의 모든 데이터를 지우고 Ubuntu를 설치하려면 선택
- Something else: 파티션을 직접 만들고 기존 OS는 유지하면서 Ubuntu를 설치하려면 선택

# 1. Linux

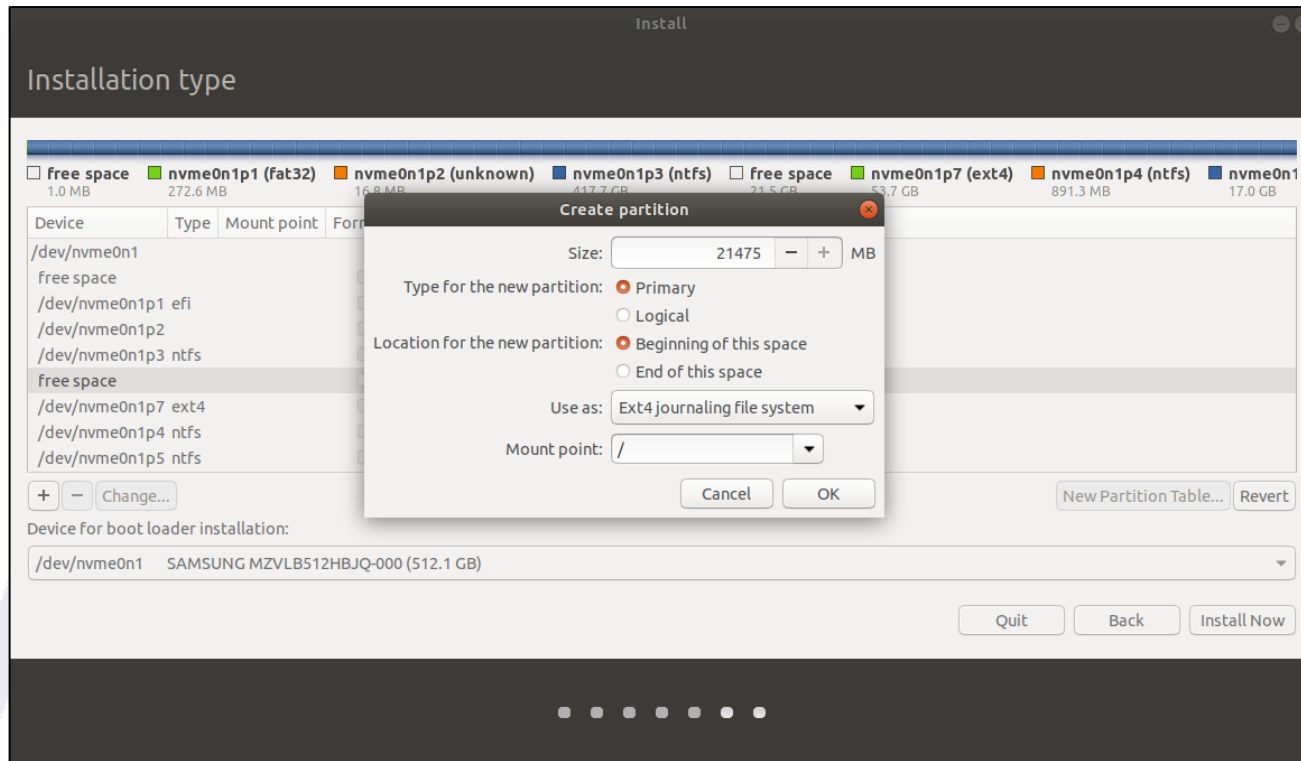
## 5) Ubuntu 18.04 LTS 설치



- 이전에 미리 분할해 놓은 파티션을 선택
- 가장 쉽게 분할된 파티션을 찾는 방법: Device 항목에서 free space를 찾고, 그 중 분할한 파티션의 크기(여기서는 20480MB)와 비슷한 크기의 free space 파티션을 선택
- 선택된 파티션을 마우스 왼쪽버튼 더블 클릭하여 create partition 창 진입

# 1. Linux

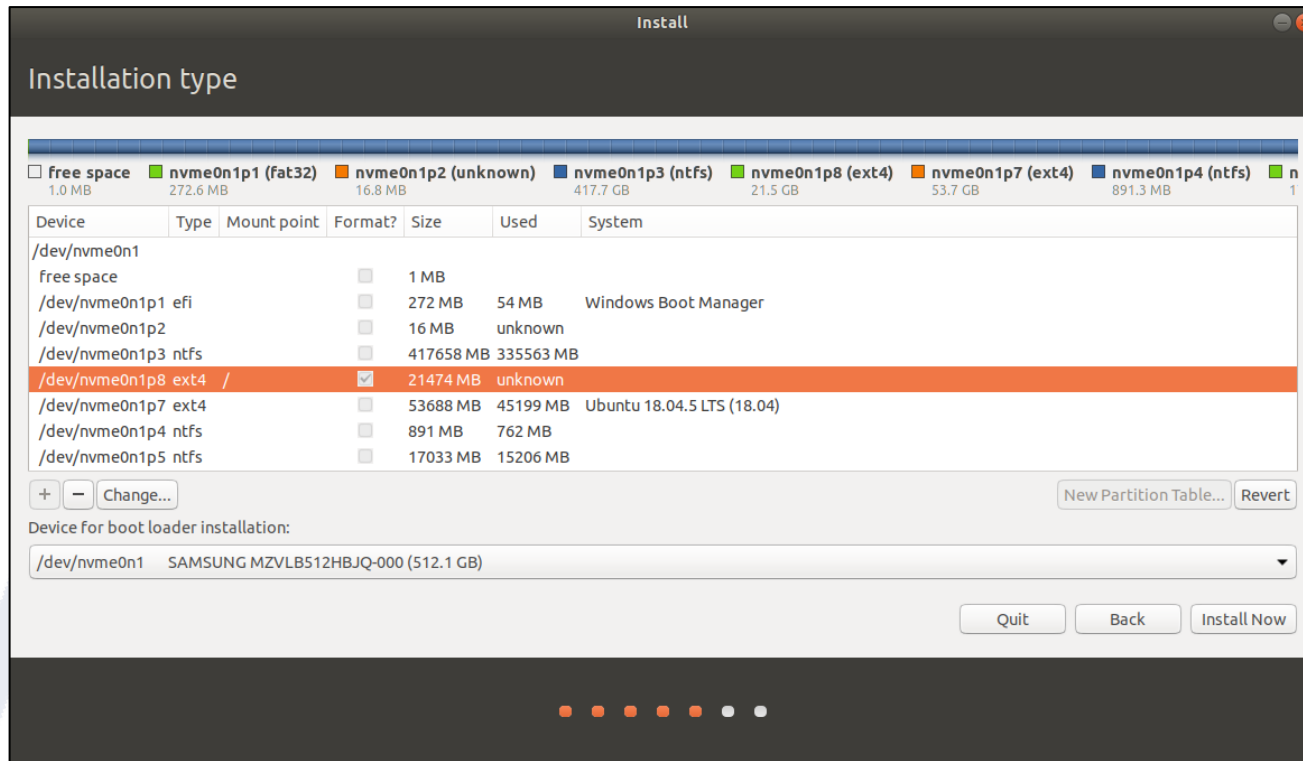
## 5) Ubuntu 18.04 LTS 설치



- 선택된 파티션을 마우스 왼쪽버튼 더블 클릭하여 Create partition 창 진입
- 다음과 같이 설정 후 **OK** 버튼 클릭
  - Size: 최대
  - Type for the new partition: **Primary**
  - Location for the new partition: **Beginning of this space**
  - Use as: **Ext4 journaling file system**
  - Mount point: **/**

# 1. Linux

## 5) Ubuntu 18.04 LTS 설치

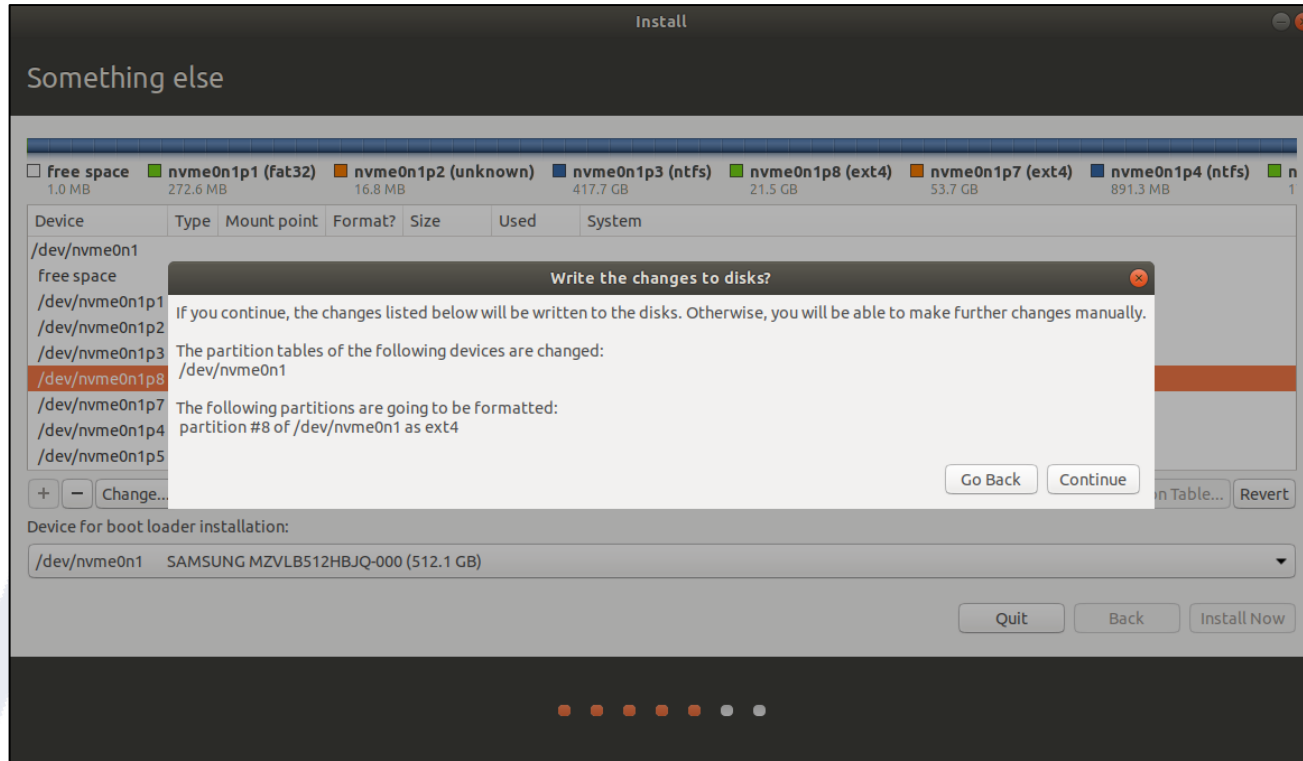


- 선택한 파티션의 Type: **ext4**, Mount point: **/**, Used: **unknown** 으로 변경되었는지 확인
- Format? 항목이 체크 되어있는지 확인 후 **Install Now** 버튼을 눌러 진행



# 1. Linux

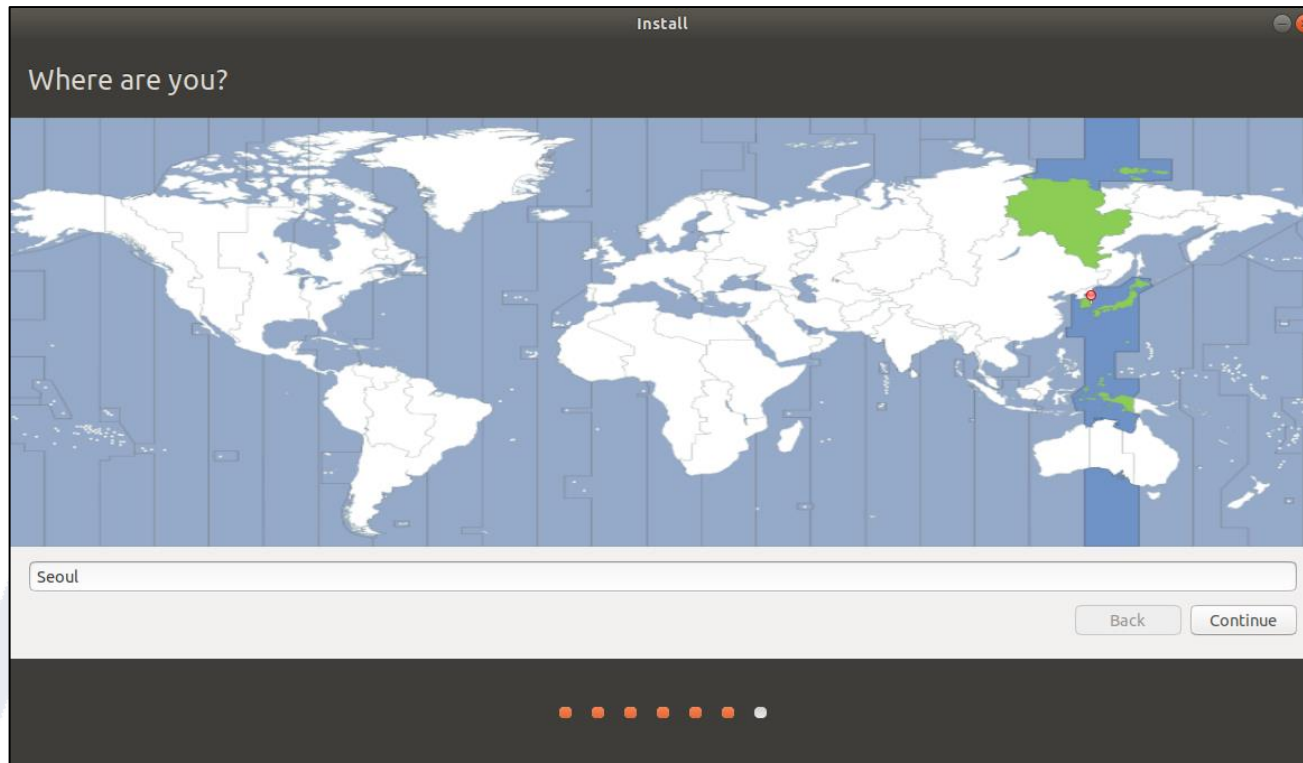
## 5) Ubuntu 18.04 LTS 설치



- 마지막으로 포맷이 실행될 디스크의 정보를 확인하고 맞다면 **Continue**를 눌러 진행

# 1. Linux

## 5) Ubuntu 18.04 LTS 설치



- 국가(시간대) 설정

# 1. Linux

## 5) Ubuntu 18.04 LTS 설치

Install

Who are you?

Your name:

Your computer's name:   
The name it uses when it talks to other computers.

Pick a username:

Choose a password:

Confirm your password:

☐ Log in automatically  
☒ Require my password to log in

Back Continue

- 사용자 정보 입력(이름, 패스워드)
- **Continue**를 누르면 Ubuntu 설치 시작

# 1. Linux

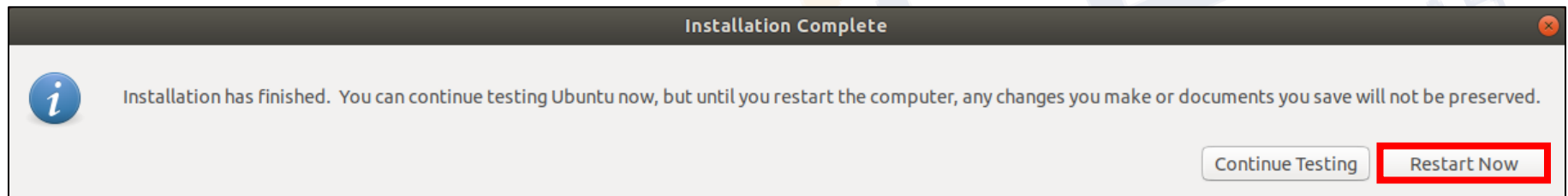
## 5) Ubuntu 18.04 LTS 설치



- 설치 진행 중 화면

# 1. Linux

## 5) Ubuntu 18.04 LTS 설치

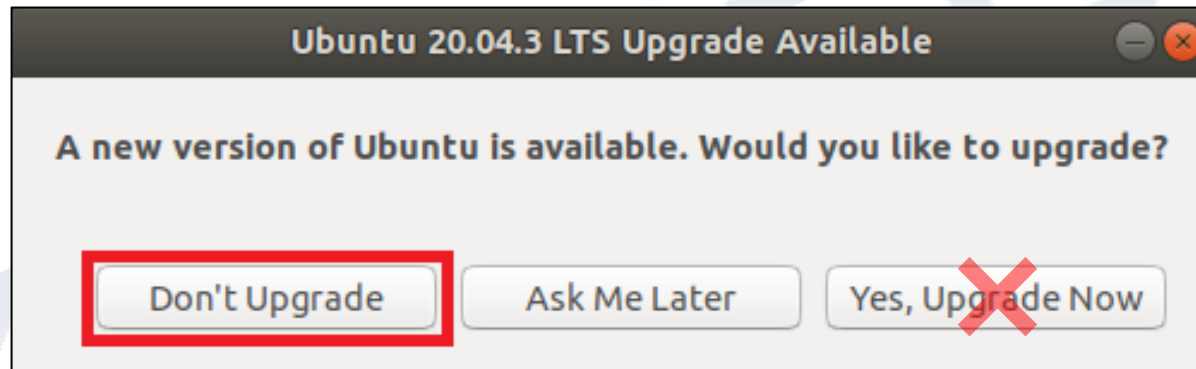


- 설치가 끝난 후 완료 문구창을 확인한 후 **Restart Now** 버튼을 눌러 재부팅
- 안내에 따라서 USB 부팅 디스크 PC로부터 제거
- 멀티 운영체제로 파티션을 나누어 설치가 되었다면, PC 부팅시에 Ubuntu로 실행할 것인지 Windows로 실행할 것인지 선택하는 창이 나타남

# 1. Linux

## 5) Ubuntu 18.04 LTS 설치

- 초기 업데이트



- 처음 Ubuntu 환경에 진입하면 그림과 같이 20.04 LTS 버전으로 업그레이드 하라는 안내창이 나타남
- 18.04 LTS 버전을 사용할 것이므로 **Don't Upgrade**를 눌러 진행
- 만약 Yes, Upgrade Now를 눌러 업그레이드를 진행하였다면, 18.04 LTS 버전을 처음부터 재설치해야함

# 1. Linux

## 5) Ubuntu 18.04 LTS 설치

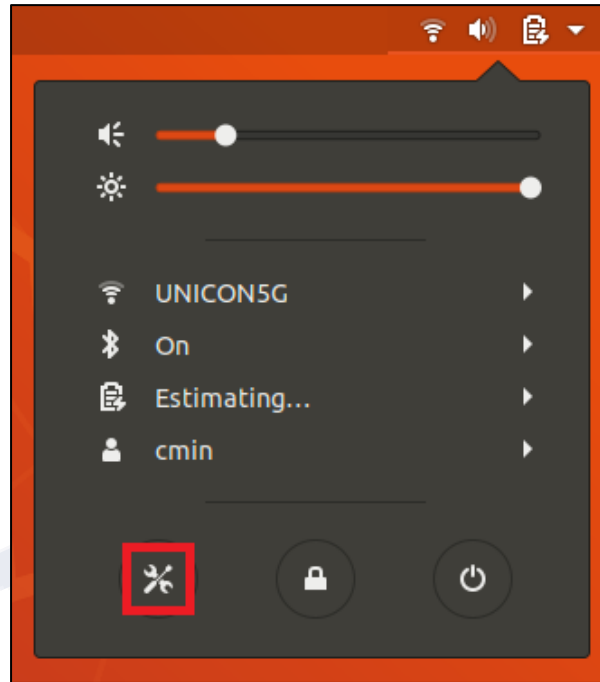
- 초기 업데이트



- 네트워크에 연결된 상태라면 Software Updater 창이 나타남
- 시스템 필수 소프트웨어 업데이트이므로 **Install Now**를 눌러 진행
- 모든 업데이트 완료 후 **Ubuntu 재부팅**

# 1. Linux

## 6) 사용 언어 및 한영 전환키 설정 (선택)

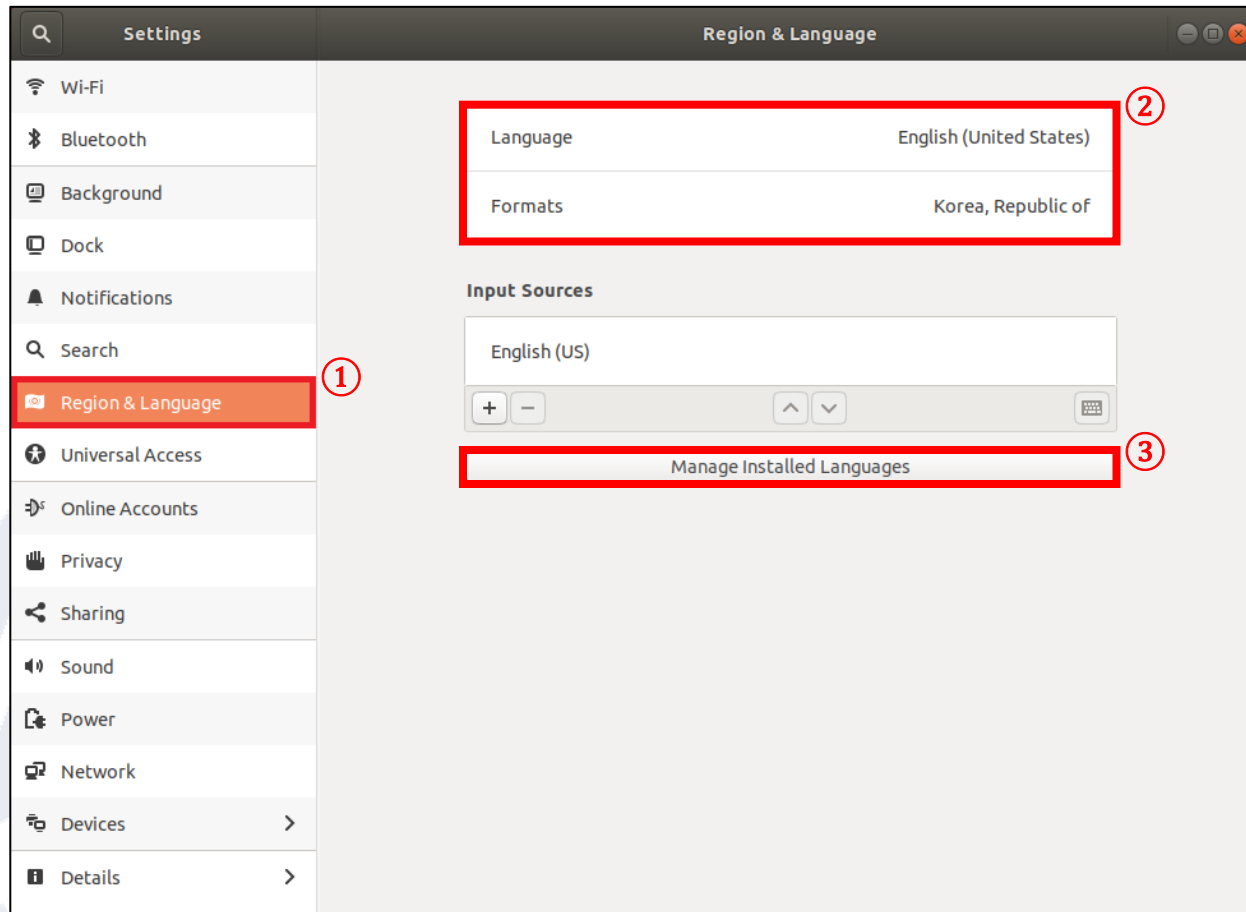


- 우측 상단의 메뉴를 눌러 **Setting** 창으로 진입



# 1. Linux

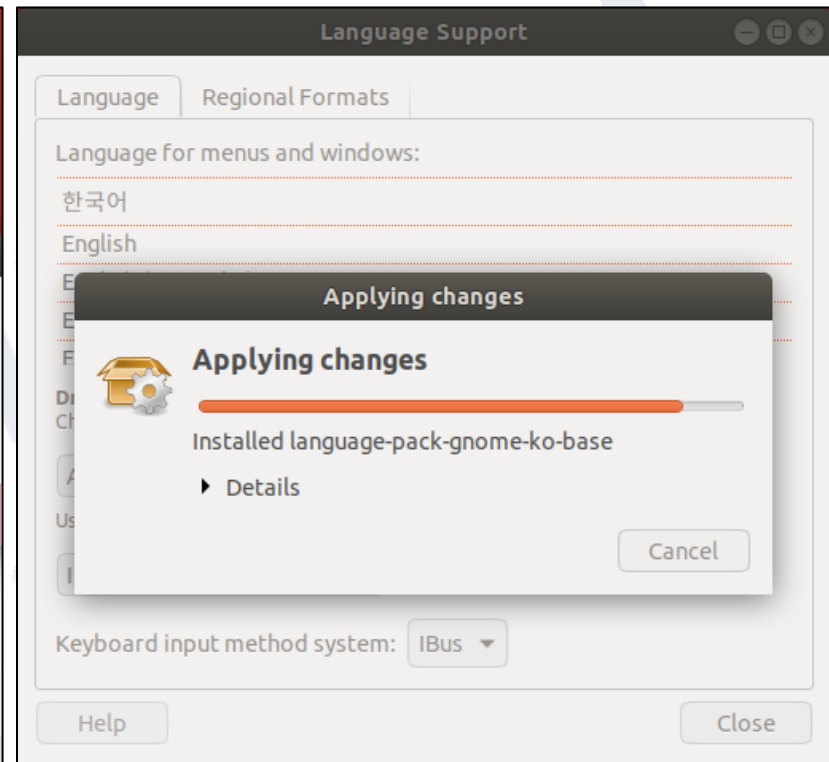
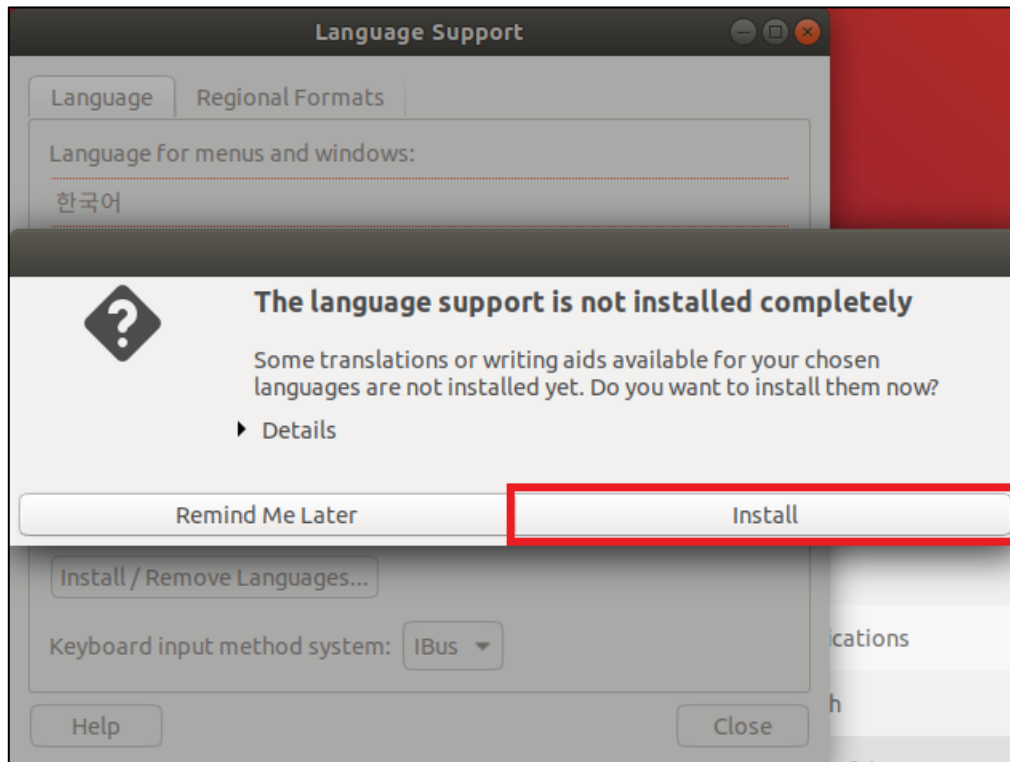
## 6) 사용 언어 및 한영 전환키 설정 (선택)



- Region & Language 탭 선택
- Language 항목에서 Korean으로 선택. Restart 버튼이 활성화되지만 지금 누르지 않음
- Manage Installed Languages를 눌러서 진입

# 1. Linux

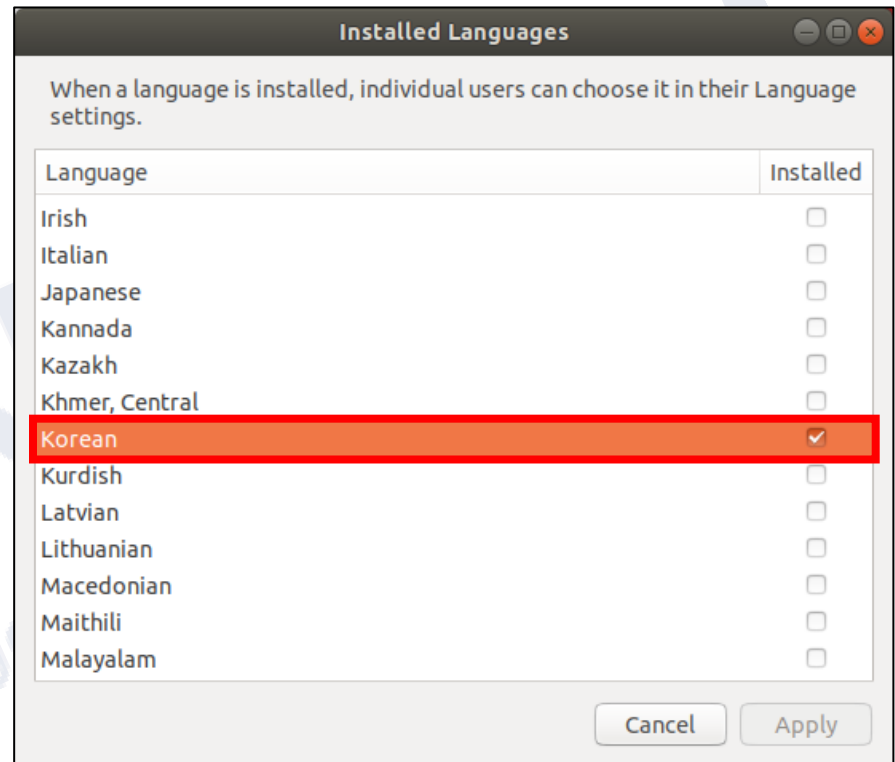
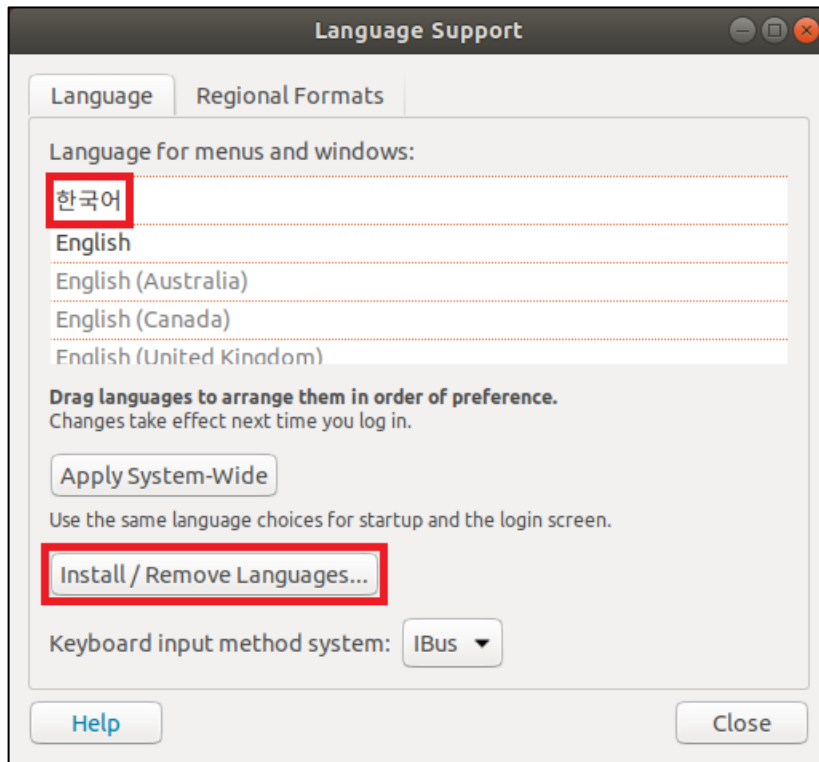
## 6) 사용 언어 및 한영 전환키 설정 (선택)



- Language Support 창에서 필요한 언어 패키지를 업데이트하기 위해 **Install**을 눌러 설치 진행

# 1. Linux

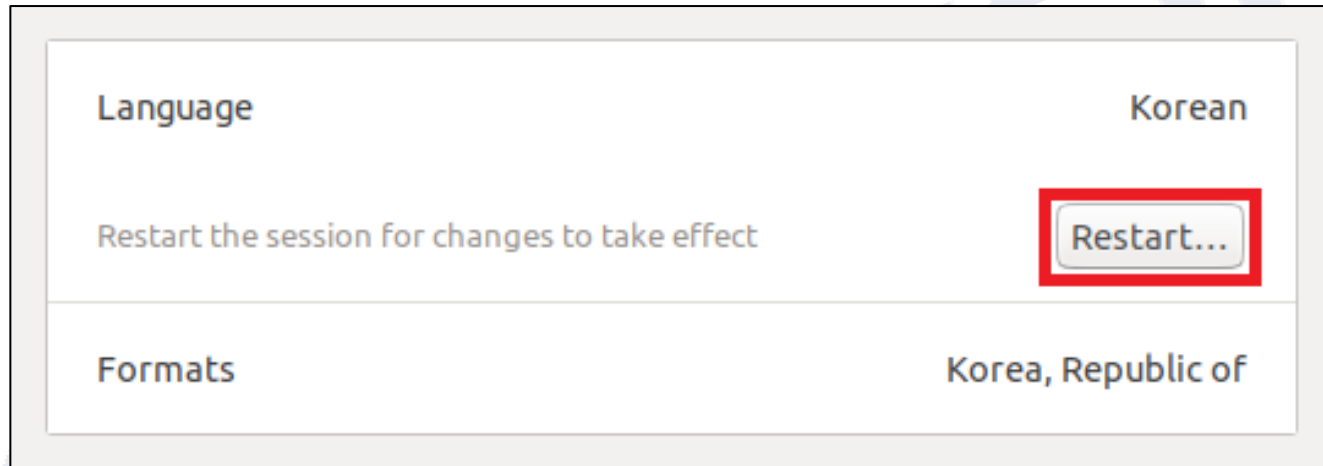
## 6) 사용 언어 및 한영 전환키 설정 (선택)



- 언어 패키지 업데이트가 완료되면 그림과 같이 Language for menus and windows 항목에 한국어가 있는지 확인
- 만약 한국어가 제대로 설치되지 않았다면, 하단의 **Install / Remove Languages**에서 **Korean**을 찾아 Installed 항목을 체크하여 한국어 언어 패키지를 설치
- 완료되면 **Apply** 및 **Close**

# 1. Linux

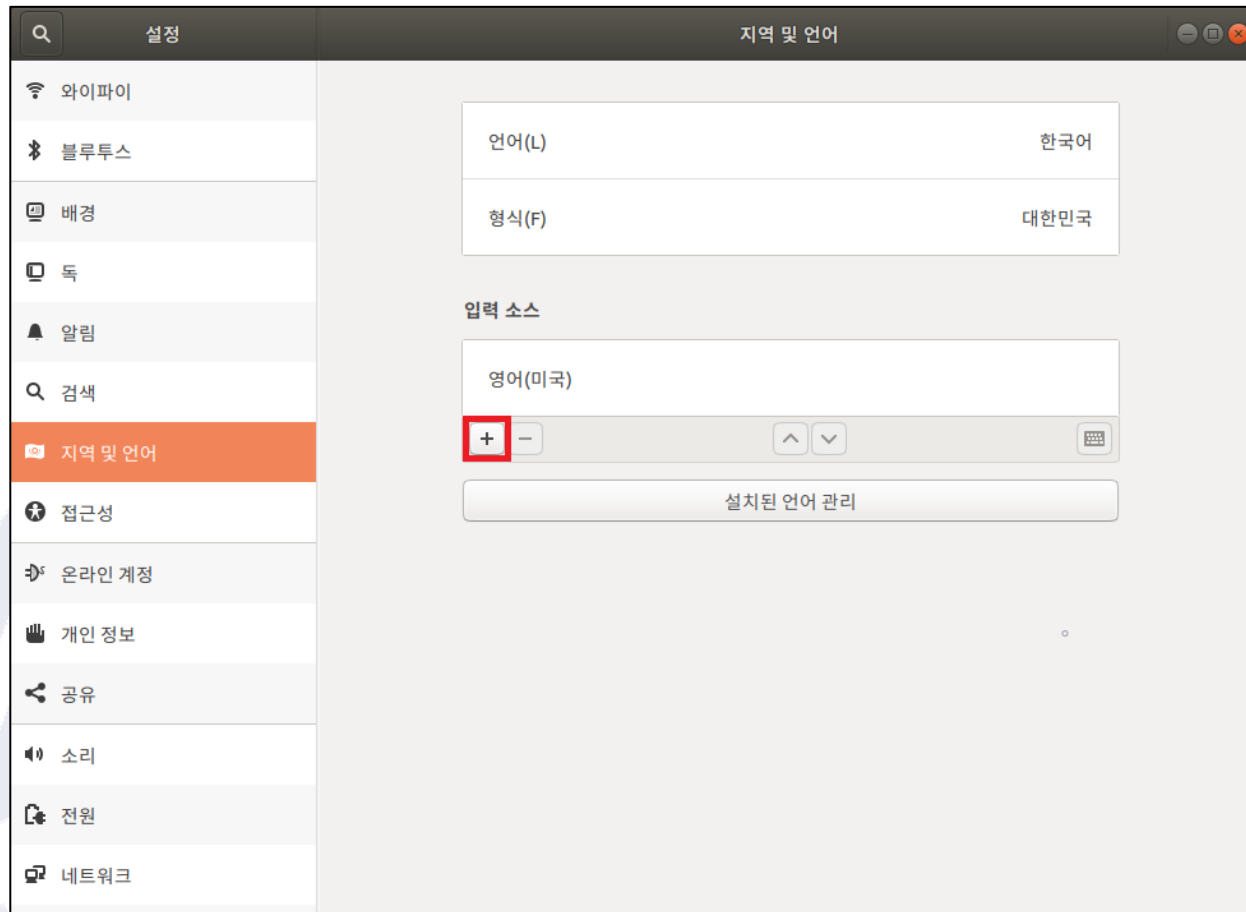
## 6) 사용 언어 및 한영 전환키 설정 (선택)



- 한국어 언어 패키지 설치가 끝났으면 언어 적용을 위해 **Restart** 버튼을 눌러 재부팅

# 1. Linux

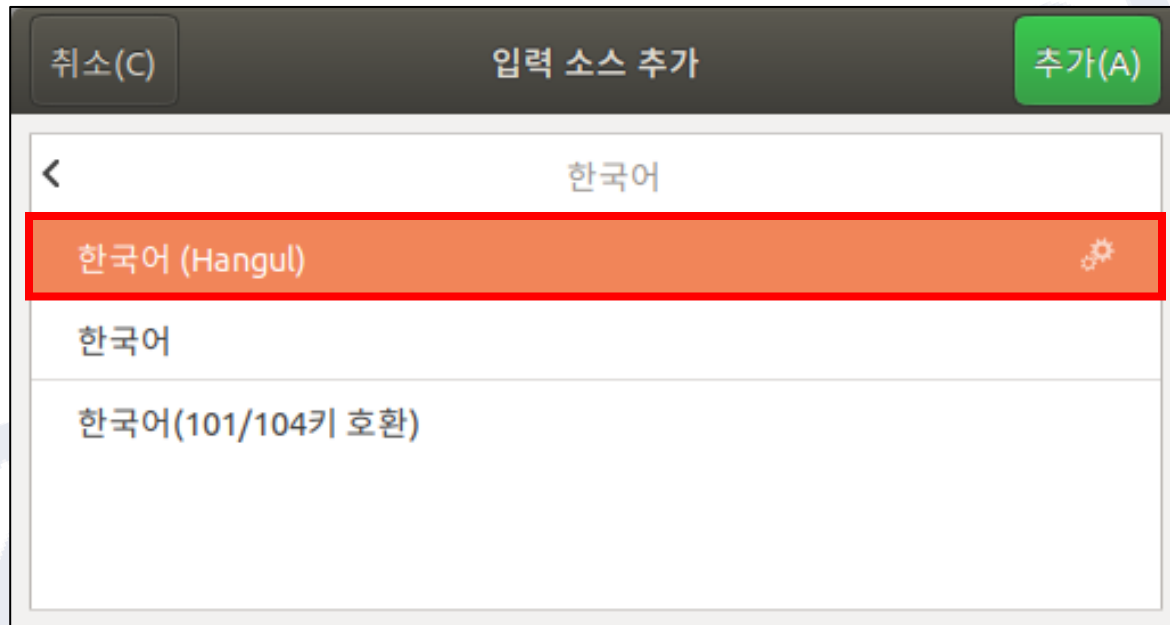
## 6) 사용 언어 및 한영 전환키 설정 (선택)



- 재부팅 후 **Region & Language** 탭 선택
- 언어 항목이 한국어로 변경된 것을 확인 후 입력 소스의 + 버튼을 눌러 입력 소스 추가

# 1. Linux

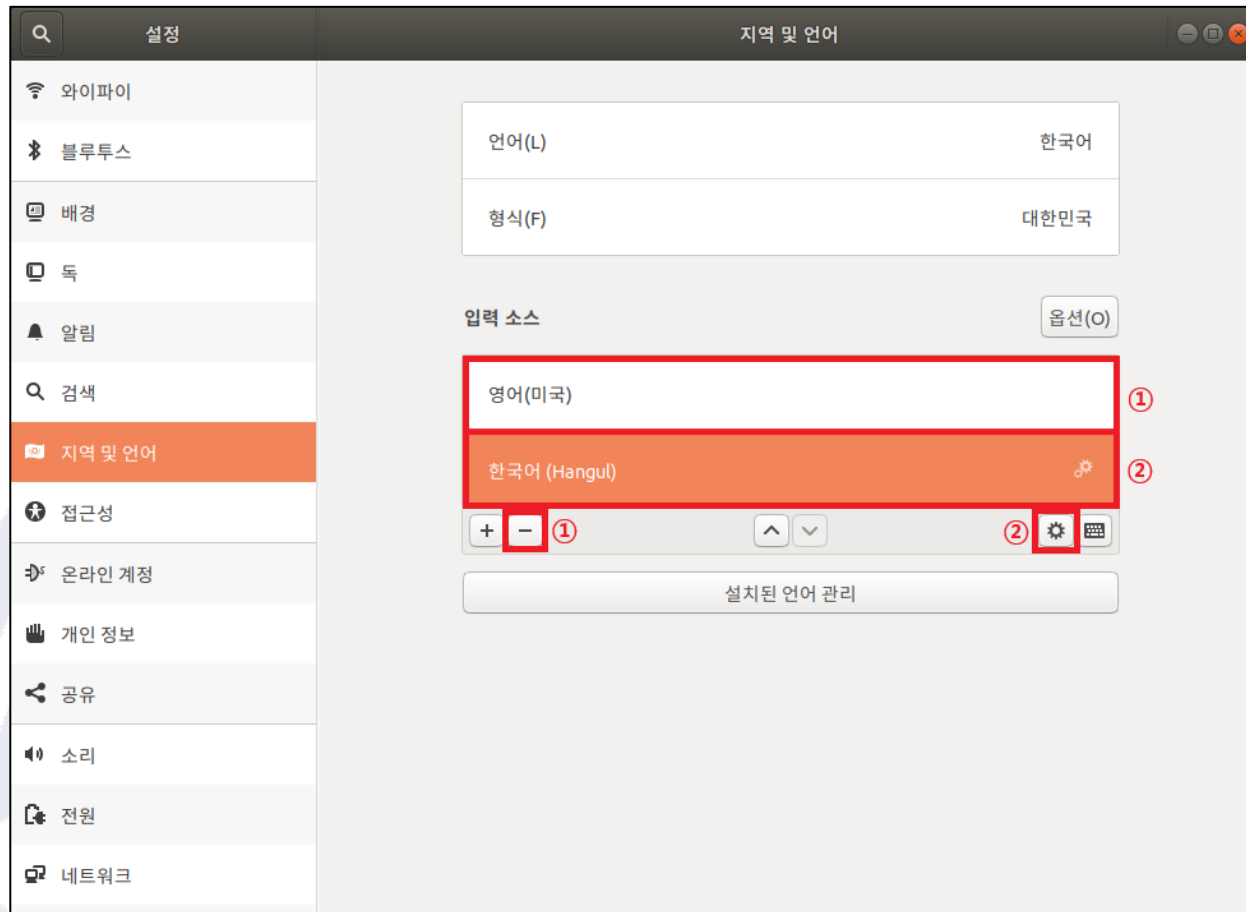
## 6) 사용 언어 및 한영 전환키 설정 (선택)



- 한국어를 선택하고, 세 종류의 항목 중 우측에 톱니바퀴 모양이 있는 **한국어 (Hangul)**을 선택하여 추가

# 1. Linux

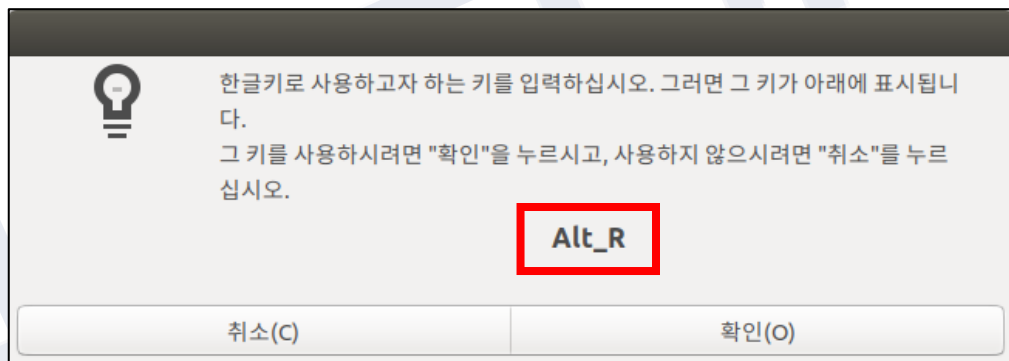
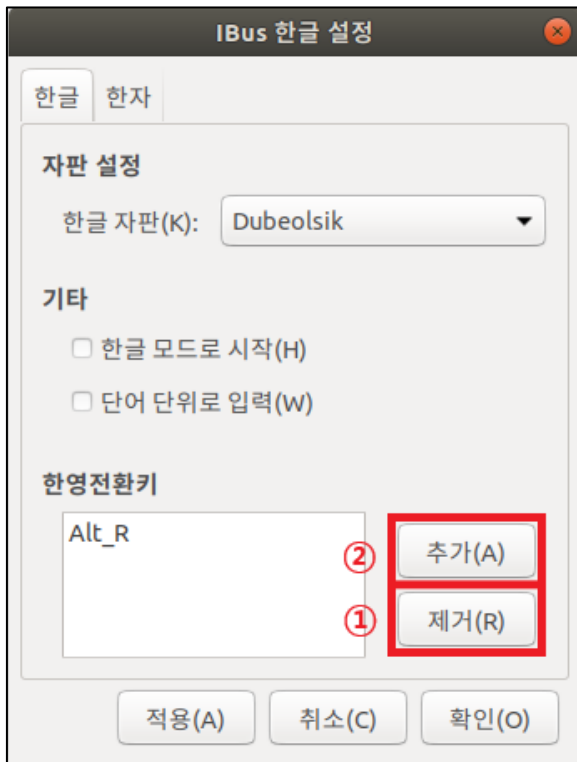
## 6) 사용 언어 및 한영 전환키 설정 (선택)



- 기본 언어로 설정된 영어(미국)을 삭제
- 한국어 (Hangul)을 선택하고 톱니바퀴 버튼을 눌러 설정창으로 진입

# 1. Linux

## 6) 사용 언어 및 한영 전환키 설정 (선택)



- IBus 한글 설정 창에서 기존의 한영전환키를 모두 제거
- 추가 버튼을 눌러 본인 키보드의 한영전환키를 눌러 키 입력
- 적용 및 확인을 눌러 설정 저장



# 1. Linux

## 7) 주요 명령어

- `ctrl + alt + t`: 터미널 실행
- `ctrl + shift + t`: 현재 터미널에서 새로운 탭으로 터미널 실행
- `ctrl + shift + w`: 탭으로 실행된 터미널 종료
- `ctrl + shift + q`: 현재 터미널 종료
- `ctrl + shift + f`: 터미널 내에서 문자열 검색
- `ctrl + shift + c`: 복사하기
- `ctrl + shift + v`: 붙여넣기
- `shift + PageUp`: 터미널 내의 상위 페이지 이동
- `shift + PageDown`: 터미널 내의 하위 페이지 이동

# 1. Linux

## ㄱ) 주요 명령어

- **(sudo) apt**: 패키지 관리 명령어. update, upgrade, install, remove, purge, policy 등 추가 명령어를 붙여 사용
- **(sudo) chmod**: 파일/폴더 권한을 설정하는 명령어. 뒤에 8진법 숫자나 옵션을 통해 권한을 부여
- **(sudo) chown**: 파일/폴더의 소유권을 설정하는 명령어. 계정명(UID), 그룹명(GID), 파일 또는 디렉터리 형태로 사용
- **ls**: 디렉토리 목록 확인
- **cd**: 디렉토리 이동
- **mkdir**: 새 디렉토리 생성
- **rm**: 파일 또는 디렉토리 삭제
- **cp**: 파일 또는 디렉토리 복사
- **mv**: 파일 또는 디렉토리 이동 (이름을 변경할 때에도 사용)
- **touch**: 비어있는 파일 생성
- **cat**: 파일 내용 보기
- **pwd**: 터미널 내 현재 위치 확인
- **clear**: 터미널 내용 삭제

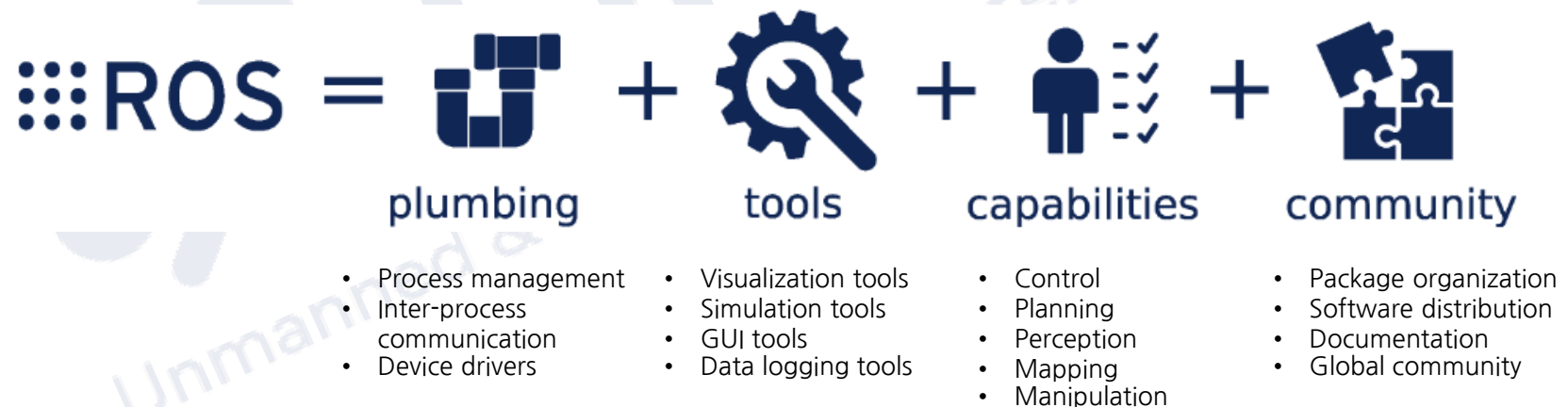
# ROS

## 2. ROS

### 1) ROS 개요

- ROS1

- ROS is an **open-source, meta-operating system** for your robot.
- Building an ecosystem to enable robotics software development to collaborate at a global level
- Make it as easy to build robot software
- To support code reuse in robotics research and development.
- **Goal: sharing and collaboration**



## 2. ROS

### 1) ROS 개요

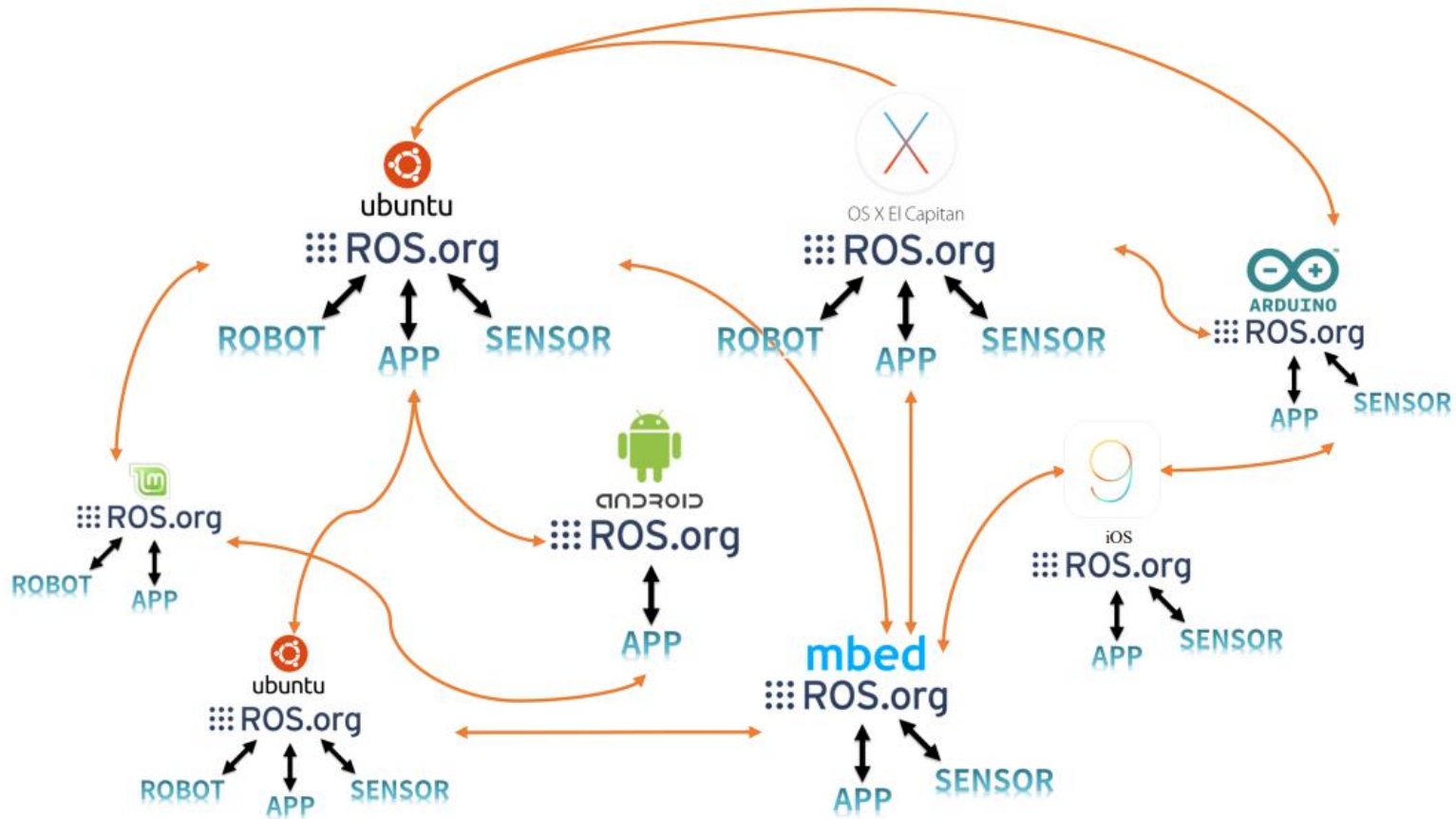
- 메타 운영체제 (Meta-Operating System)



## 2. ROS

### 1) ROS 개요

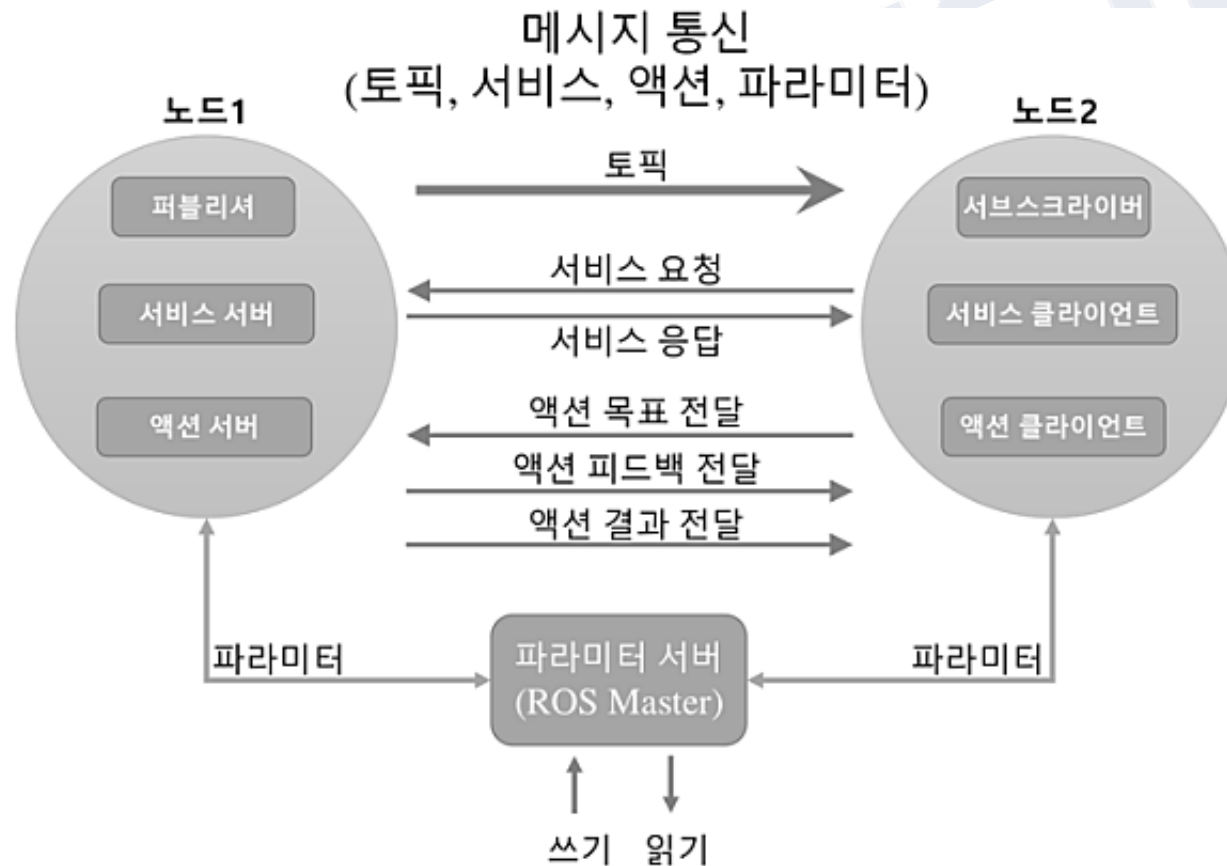
- 다양한 이기종 다바이스 간의 통신 지원



## 2. ROS

### 1) ROS 개요

- ROS1 통신 방식



## 2. ROS

### 1) ROS 개요

- ROS1의 문제점?

- Multiple robots
- Small embedded platforms
- Cross platform support
- Real-time systems
- Non-ideal networks
- Safety
- Security
- Production environments
- Prescribed patterns for building and structuring systems

API changes

Risk associated with changing the current ROS system



## 2. ROS

### 1) ROS 개요

- ROS2

- DDS
- Support for small embedded system
- Real-time control & deterministic execution & real-time communication
- Shorten time to market
- Designed for production
- Multi-platform
- Multi-domain
- No vendor lock-in
- Built on open standards
- Permissive open source license
- Global community
- Industry support
- Interoperability with ROS1

※ DDS

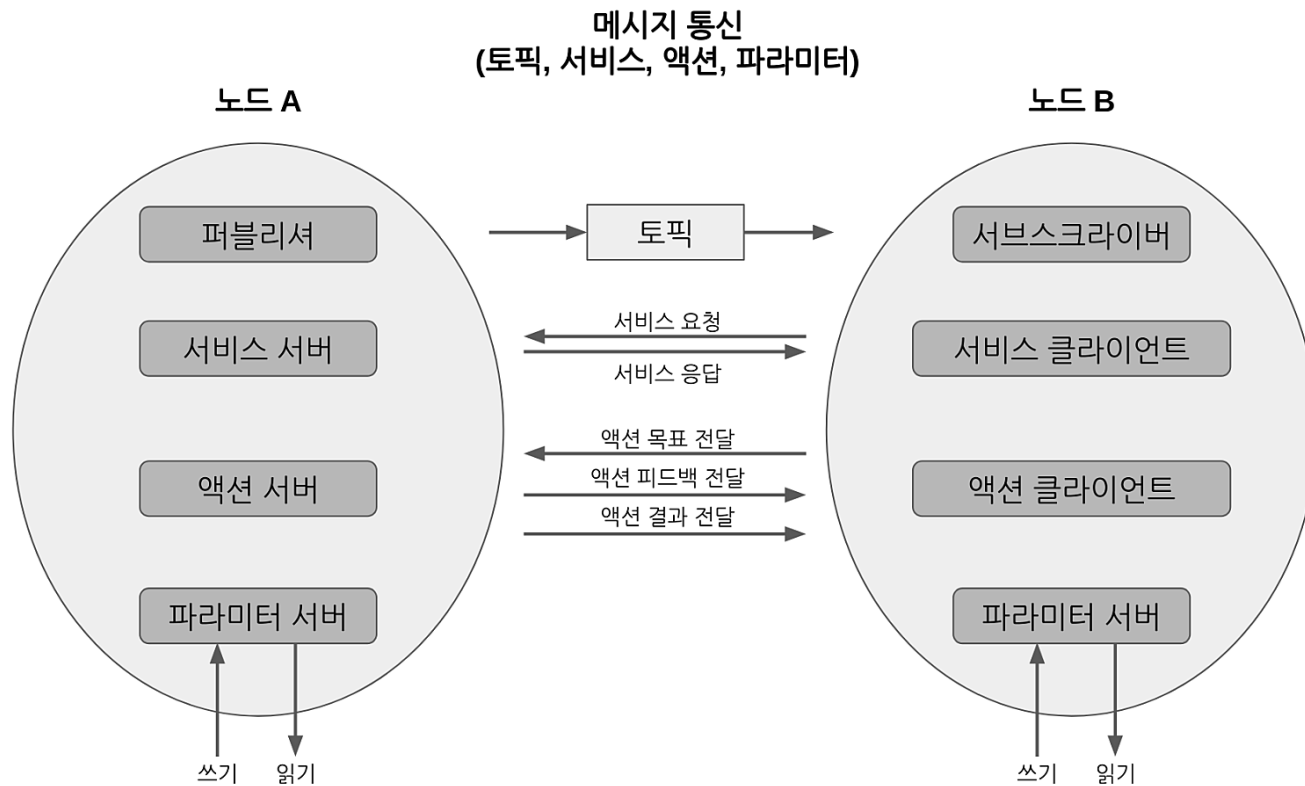
OMG(Object Management Group)에서 표준화한 실시간 publisher-subscriber 방식 통신 미들웨어



## 2. ROS

### 1) ROS 개요

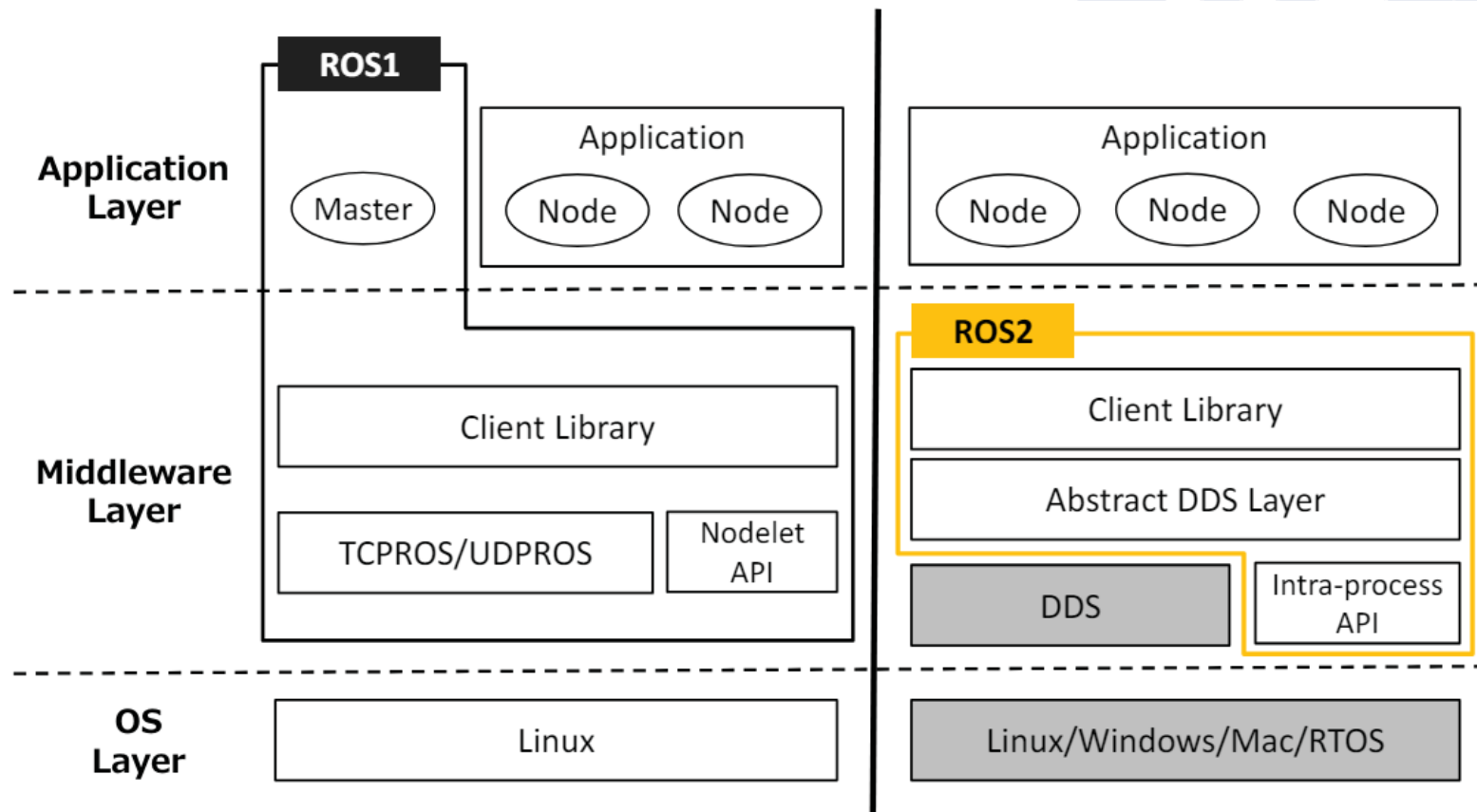
- ROS2 통신 방식



## 2. ROS

### 1) ROS 개요

- ROS1 vs ROS2



## 2. ROS

### 1) ROS 개요

- ROS1 vs ROS2

Features	ROS 1	ROS 2
Platforms	Linux	Linux, macOS, Windows
Languages	C++03, Python 2.7	C++14 (C++17), Python 3.5+
Communication	XMLRPC + TCPROS	DDS (RTPS)
Node Manager	ROS Master	No, use DDS's auto-discovery
Build System	roscpp → catkin (CMake)	ament (CMake), Python setuptools (Full support)
Build Tools	catkin_make, catkin_tools	colcon
Build Options	-	Multiple workspace, No non-isolated build, No devel space
Version Control System	roscpp → wstool, roscpp (*rosinstall)	vcstool (*repos)
Multiple Nodes	one node in a process	multiple nodes in a process
Messages	*.msg, *.srv, *.action	*.msg, *.srv, *.action, *.idl
roslaunch	XML	Python, XML, YAML
Security	SROS	SROS 2, DDS-Security, Robotic Systems Threat Model
Real-time	external frameworks like OROCOS	real-time nodes when using a proper RTOS with carefully written user code
Graph API	remapping at startup time only	remapping at runtime
Embedded Systems	roscpp, mROS	microROS, ros2arduino, H-ROS, Renesas DDS-XRCE(Micro-XRCE-DDS), AWS ARCLM

## 2. ROS

### 1) ROS 개요

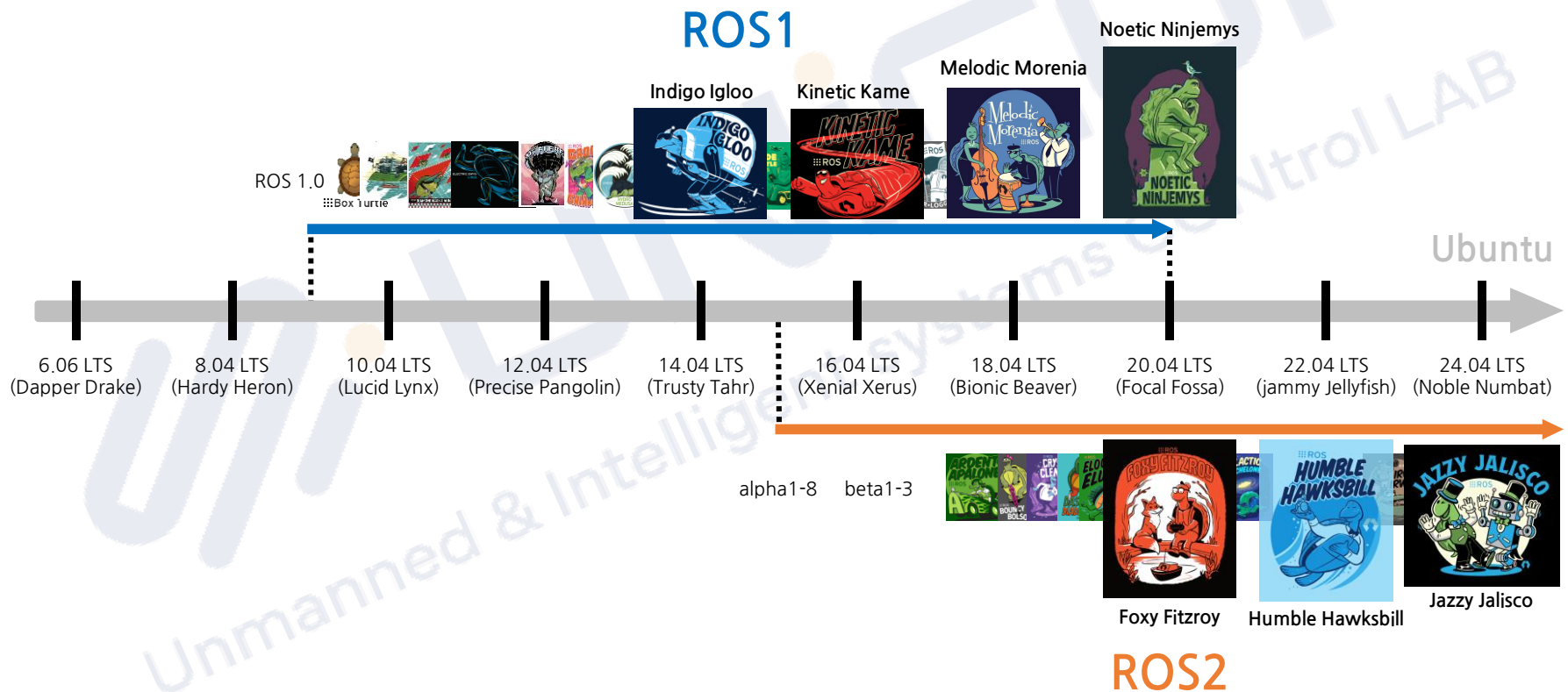
- ROS1 vs ROS2

ROS1	<ul style="list-style-type: none"> <li>• 단일 로봇</li> <li>• 워크스테이션급 컴퓨터</li> <li>• Linux 환경</li> <li>• 실시간 제어 지원하지 않음</li> <li>• 안정된 네트워크 환경이 요구됨</li> <li>• 주로 대학이나 연구소와 같은 아카데믹 연구 용도</li> </ul>
ROS2	<ul style="list-style-type: none"> <li>• 복수대의 로봇</li> <li>• 임베디드 시스템에서의 ROS 사용</li> <li>• 실시간 제어</li> <li>• 불안정한 네트워크 환경에서도 동작할 수 있는 유연함</li> <li>• 멀티 플랫폼 (Linux, macOS, Windows)</li> <li>• 최신 기술 지원 (Zeroconf, Protocol Buffers, ZeroMQ, WebSockets, DDS 등)</li> <li>• 상업용 제품 지원</li> </ul>

# 2. ROS

## 1) ROS 개요

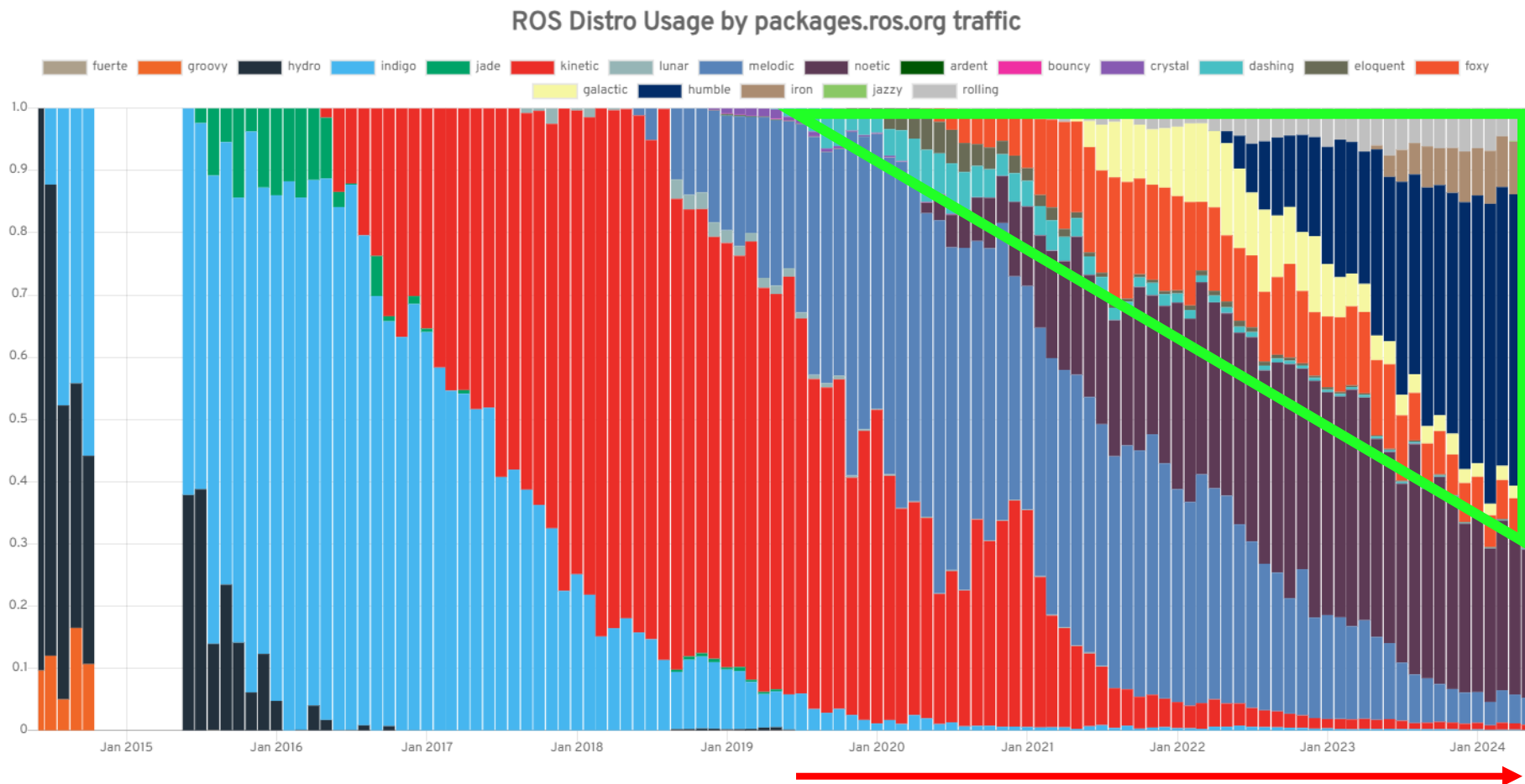
- ROS1 vs ROS2



## 2. ROS

### 1) ROS 개요

- ROS1 vs ROS2



ROS2 사용률 증가

## 2. ROS

### 1) ROS 개요

- Why ROS1?

- 입문자용으로 적합. ROS2 보다는 ROS1이 상대적으로 진입장벽이 낮음
- ROS1을 경험해보고 ROS2로 넘어가면 ROS의 전체적인 구조에 대한 이해도를 훨씬 크게 가져갈 수 있음
- 아직 ROS1 기반으로 개발하는 프로젝트가 많음
- 수많은 공식 및 비공식 패키지, 연습 예제 등



Ubuntu 18.04 LTS



ROS1 Melodic Morenia



# 2. ROS

## 1) ROS 개요

- ROS wiki 활용

[ROS wiki link](#)



### 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](#) | [العربية](#)

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.

Robots/Hardware:

#### Robots

Robots that you can use with ROS.

#### Sensors

Sensor drivers for ROS.

#### Motors

Motor controller drivers for ROS.

>> ROS 2 Documentation: <https://docs.ros.org/en/jazzy/index.html>

## 2. ROS

### 2) ROS Melodic 설치



- Ubuntu install of ROS Melodic

[ROS wiki melodic installation link](#)

#### 1.2 Setup your sources.list

packages.ros.org의 소프트웨어를 허용하도록 설정

```
$ sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -sc) main" > /etc/apt/sources.list.d/ros-latest.list'
```

#### 1.3 Set up your keys

Key를 추가하기 위한 curl이 설치되어 있지 않다면 설치

```
$ sudo apt install curl
```

다음 명령어를 통해 key를 추가

```
$ curl -s https://raw.githubusercontent.com/ros/rosdistro/master/ros.asc | sudo apt-key add -
```

## 2. ROS

### 2) ROS Melodic 설치



- Ubuntu install of ROS Melodic

[ROS wiki melodic installation link](#)

#### 1.4 Installation

Debian 패키지 목록을 최신화

```
$ sudo apt update
```

모든 ROS 툴을 포함하고 있는 full 패키지로 설치(추천). 용량의 문제가 있다면 최소 버전으로 설치(ROS wiki 참조)

```
$ sudo apt install ros-melodic-desktop-full
```

#### 1.5 Environment setup

환경변수를 추가하기 위해서는 터미널을 새로 열 때마다 source 명령어를 통해 불러와야 하지만 bashrc에 환경변수가 추가되어 있다면 매번 명령어를 입력하는 번거로움 없이 자동으로 불러올 수 있음

다음 명령어를 통해 bashrc에 ROS 환경변수를 추가

```
$ echo "source /opt/ros/melodic/setup.bash" >> ~/.bashrc
```

수정된 bashrc 적용

```
$ source ~/.bashrc
```

## 2. ROS

### 2) ROS Melodic 설치



- Ubuntu install of ROS Melodic

[ROS wiki melodic installation link](#)

#### 1.6 Dependencies for building packages

ROS 패키지 구축을 위한 도구와 기타 종속성 설치

```
$ sudo apt install python-rosdep python-rosinstall python-rosinstall-generator python-wstool build-essential
```

##### 1.6.1 Initialize rosdep

rosdep은 컴파일하려는 소스에 대한 시스템 종속성을 쉽게 설치할 수 있게 해주며, ROS에서 일부 핵심 구성 요소를 실행하는데 필요. rosdep이 설치되지 않았다면 설치

```
$ sudo apt install python-rosdep
```

다음 명령어를 통해 rosdep 초기화

```
$ sudo rosdep init
```

```
$ rosdep update
```

## 2. ROS

### 2) ROS Melodic 설치

- Turtlesim을 활용한 ROS 동작 테스트

Terminal 1

```
$ roscore
```

Terminal 2

```
$ rosrun turtlesim turtlesim_node
```

Terminal 3

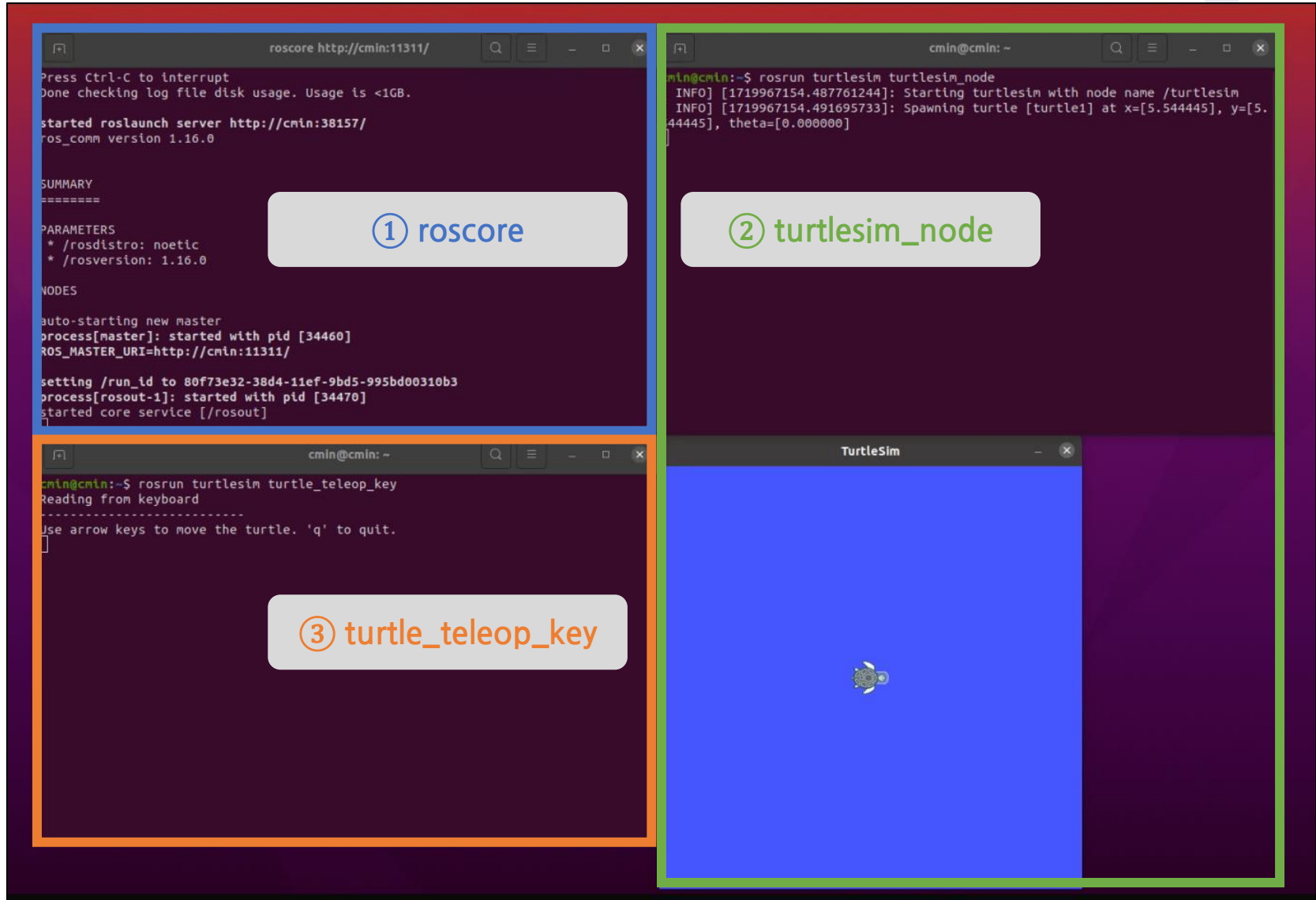
```
$ rosrun turtlesim turtle_teleop_key
```



키보드의 방향키를 이용하여 거북이를 움직이고, 오류 없이 잘 동작된다면 ROS Melodic 버전이 잘 설치되었다는 것을 의미

# 2. ROS

## 2) ROS Melodic 설치



The image displays three terminal windows and a TurtleSim window, illustrating the steps to run ROS Melodic on a Raspberry Pi.

**① roscore**

```

roscore http://cmin:11311/

Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://cmin:38157/
ros_comm version 1.16.0

SUMMARY
=====
PARAMETERS
 * /rostdistro: noetic
 * /rosversion: 1.16.0

NODES

auto-starting new master
process[master]: started with pid [34460]
ROS_MASTER_URI=http://cmin:11311/

setting /run_id to 80f73e32-38d4-11ef-9bd5-995bd00310b3
process[roscout-1]: started with pid [34470]
started core service [/roscout]
  
```

**② turtlesim\_node**

```

cmin@cmin: ~
cmin@cmin:~$ rosrn turtlesim turtlesim_node
INFO] [1719967154.487761244]: Starting turtlesim with node name /turtlesim
INFO] [1719967154.491695733]: Spawning turtle [turtle1] at x=[5.544445], y=[5.44445], theta=[0.000000]
  
```

**③ turtle\_teleop\_key**

```

cmin@cmin: ~
cmin@cmin:~$ rosrn turtlesin turtle_teleop_key
Reading from keyboard
-----
Use arrow keys to move the turtle. 'q' to quit.
  
```

**TurtleSim**

The TurtleSim window shows a blue background with a small turtle icon in the center, representing the simulated environment.

## 2. ROS

### 3) ROS1 기본 용어

- Master

- 노드 간 연결 및 메시지 통신을 위한 네임 서버 역할
- 마스터를 실행하면 각 노드의 이름을 등록하고 필요할 때 정보를 얻음
- 노드 간 연결과 토픽, 서비스 등 메시지 통신은 마스터 없이는 불가능

- Node

- ROS에서 실행되는 가장 작은 단위의 프로세서
- 예를 들어 이동 로봇의 경우, 센서 구동, 센서 데이터 변환, 장애물 인식, 모터 구동, 네비게이션 등의 기능별로 노드를 나눔

- Message

- 메시지를 통해 노드 간에 데이터를 송수신함
- Float, int, bool, string 등의 변수

## 2. ROS

### 3) ROS1 기본 용어

- Topic

- ROS에서 가장 많이 사용하는 메시지의 종류
- Publisher 노드는 자신의 토픽을 마스터에 등록한 후 해당 토픽에 대한 메시지를 송신
- Subscriber 노드는 수신하고자 하는 메시지를 송신하는 publisher에 대한 정보를 마스터에 요청하고, 이를 바탕으로 publisher 노드와 연결하여 토픽으로 메시지를 교환

- Publisher

- 토픽 메시지를 송신하는 것을 publish라 하며, 그 역할을 publisher가 수행
- 한 노드에서 여러 publisher가 선언될 수 있음

- Subscriber

- 토픽 메시지를 수신하는 것을 subscribe라 하며, 그 역할을 subscriber가 수행
- 한 노드에서 여러 subscriber가 선언될 수 있음



## 2. ROS

### 4) ROS1 기본 명령어

- **roscore**
  - ROS 마스터를 실행하는 명령어
  - 한 네트워크 환경 내에서 하나의 roscore만 실행되어야 함
- **roslaunch [패키지명] [노드명]**
  - ROS 패키지의 단일 노드를 실행시키는 명령어
- **roslaunch [패키지명] [런치파일명]**
  - 여러 노드를 한번에 실행시키는 명령어
  - 패키지 파라미터나 노드 이름 변경, 노드의 네임스페이스 구성, 환경 변수 설정 등의 부가 기능을 갖춘 노드 실행에 특화
  - '\*.launch' 파일을 따로 작성하여 구성

## 2. ROS

#### 4) ROS1 기본 명령어

- roscd [패키지명]: 지정한 패키지의 디렉토리로 이동
  - rosls [패키지명]: 지정한 패키지의 파일 목록 확인
  - rospack find [패키지명]: 설치된 패키지의 위치를 확인
- 
- rosnode list: 활성화되어 있는 노드 목록 확인
  - rosnode info [노드명]: 지정된 노드의 정보 확인
  - rosnode kill [노드명]: 지정된 노드 종료
  - rosnode cleanup: 연결이 끊어진 유령 노드의 등록 정보 삭제
- 
- rostopic list: 활성화되어 있는 토픽 목록 확인
  - rostopic echo [토픽명]: 지정된 토픽의 메시지 내용을 실시간으로 표시
  - rostopic type [토픽명]: 지정된 토픽의 메시지 타입 확인
  - rostopic hz [토픽명]: 지정된 토픽의 publish 주기를 확인
  - rostopic info [토픽명]: 지정된 토픽의 정보 확인
  - rostopic pub [토픽명] [메세지타입] [값]: 지정된 토픽의 형태로 메시지 publish

## 2. ROS

### 5) 개발 환경 구축

- IDE (Integrated Development Environment)

#### VS code (Visual Studio Code) 추천

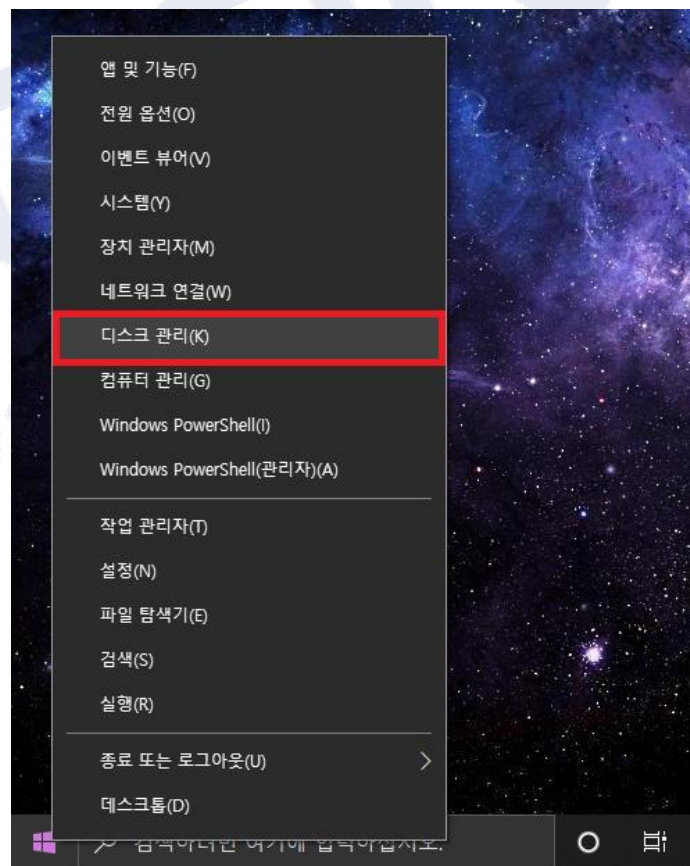
- linux, windows, macOS 모두 사용 가능한 크로스 플랫폼 에디터
- Extension을 통한 다양한 프로그래밍 언어 지원
- **Install:** <https://code.visualstudio.com/>
- **Extension:** C/C++, Cmake Tools, Python, ROS, URDF, Visual Studio IntelliCode, Project Manager, Code runner, Material Theme, ...

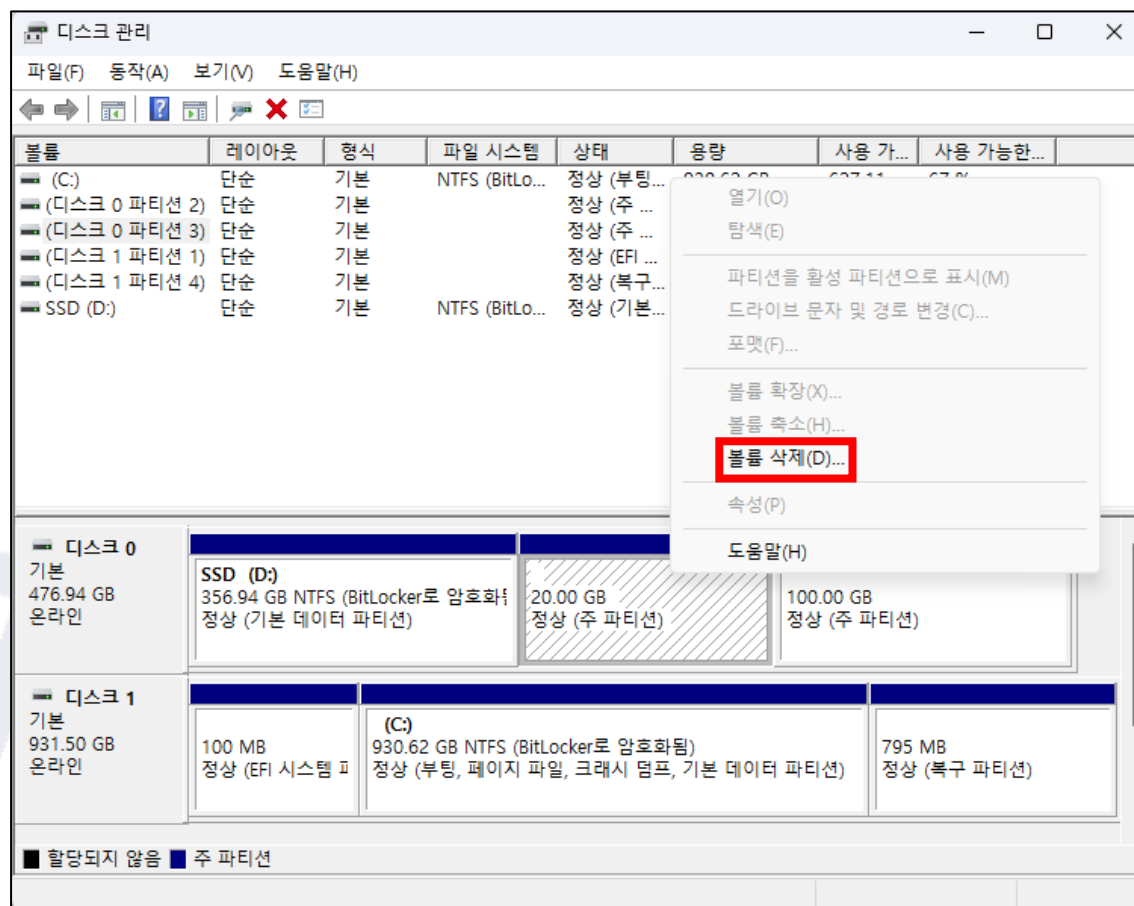


# Appendix

※ 현재 ubuntu 부팅으로 설정되어 있다면, 반드시 메인보드의 bios 세팅으로 진입해 boot option의 디스크 부팅 우선순위를 ubuntu가 아닌 windows boot manager로 변경하고 시작할 것

- Window로 부팅
- 시작메뉴를 오른쪽 클릭하여 디스크 관리 창으로 진입

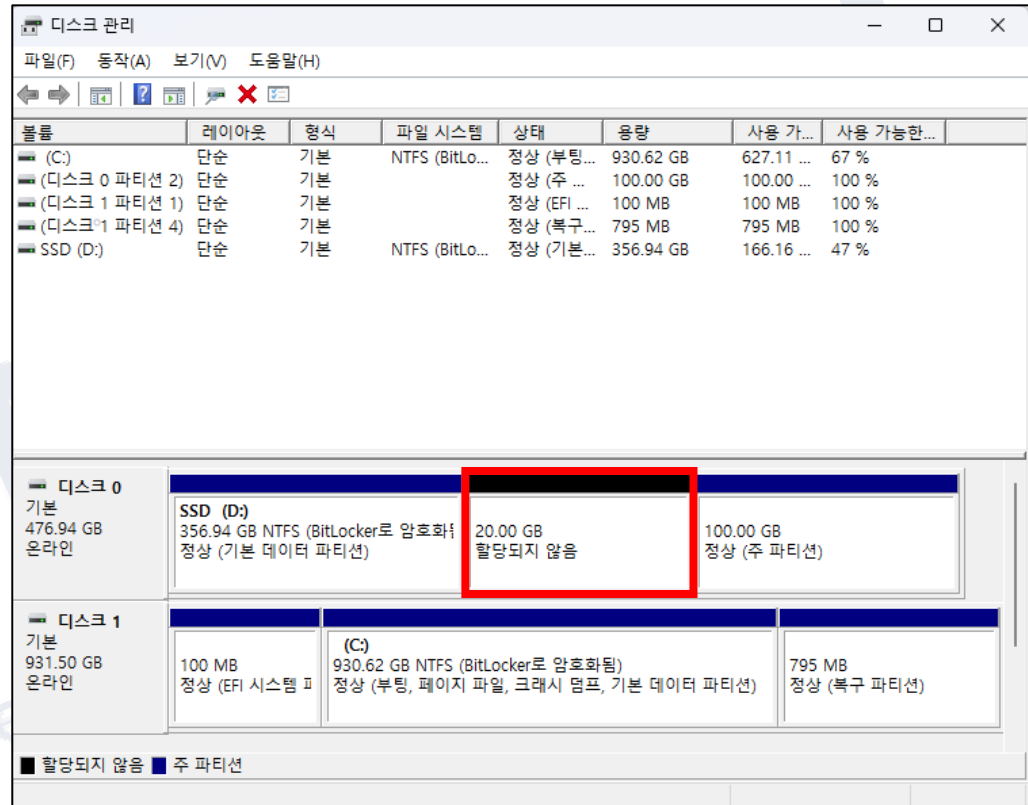
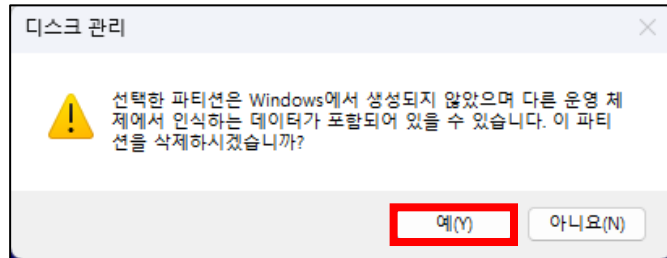




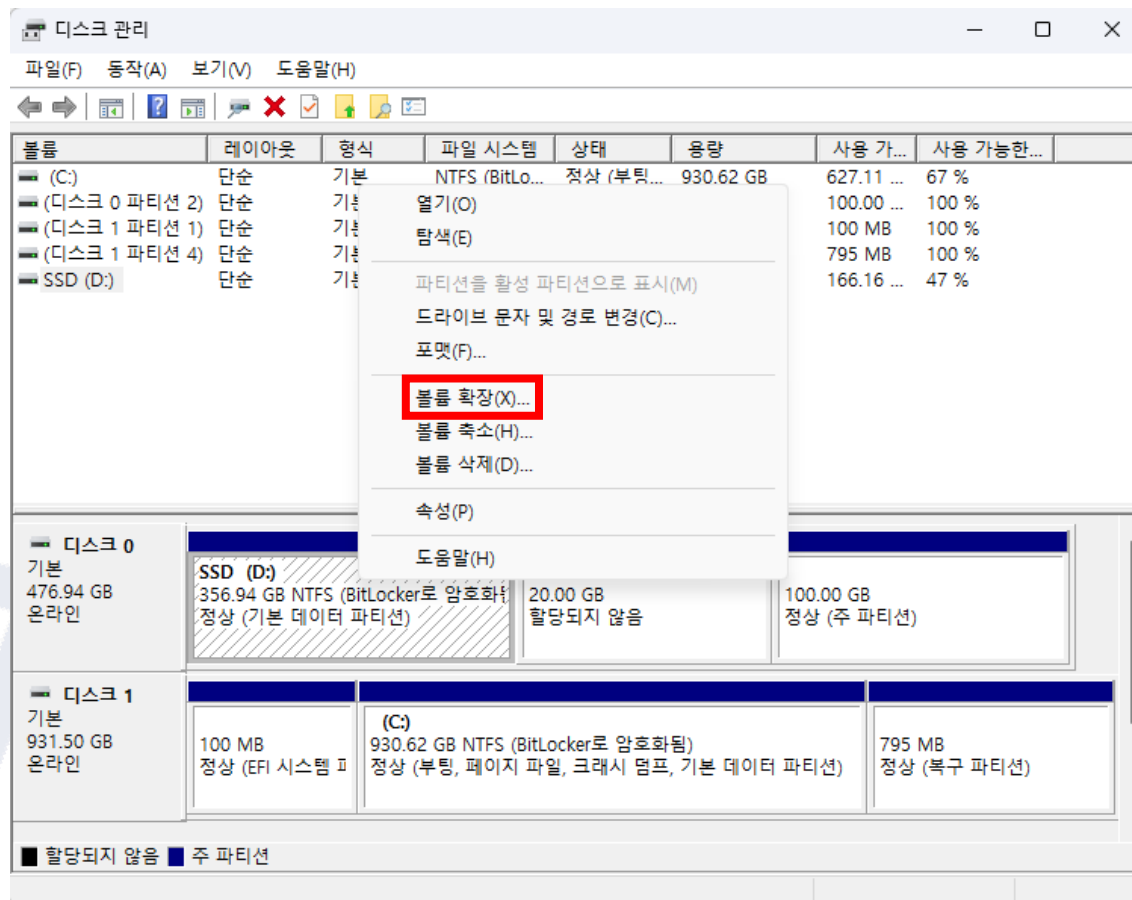
- 삭제하고자 하는 ubuntu 파티션을 오른쪽 클릭
- **볼륨 삭제** 선택 (ubuntu의 모든 내용 삭제)

# Appendix

## Ubuntu 파티션 삭제 방법



- 경고문이 나타났을 때 확인 후 **예**를 선택
- 볼륨이 삭제되면 **활당되지 않음** 상태로 변함

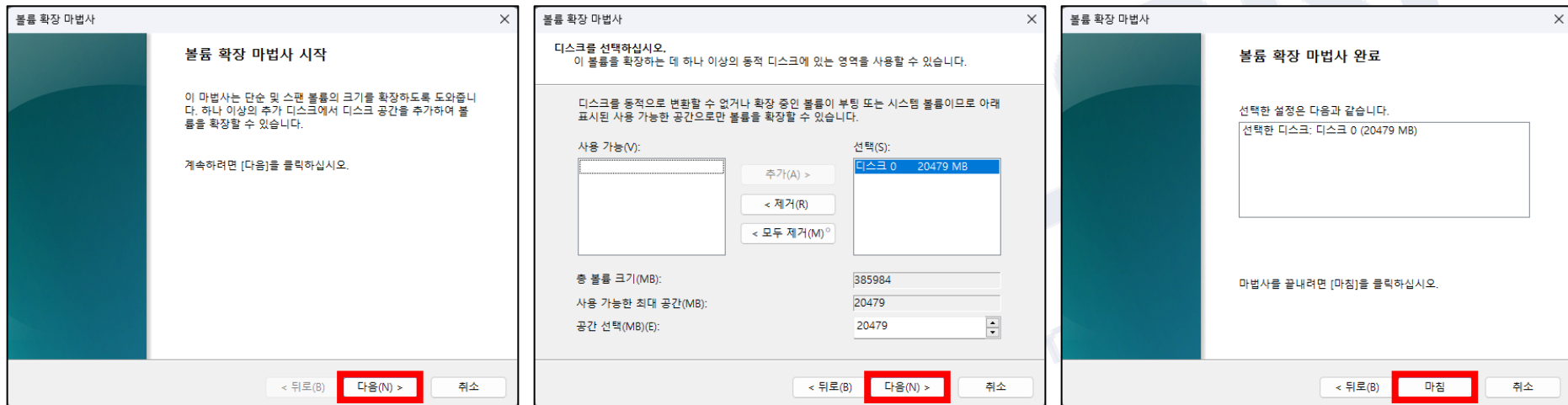


- 할당되지 않은 파티션을 합치고자 하는 주 파티션에 오른쪽 클릭
- 볼륨 확장 선택



# Appendix

## Ubuntu 파티션 삭제 방법



- 볼륨 확장 마법사 창이 나타나면 내용을 확인 후 마지막까지 다음을 눌러 진행
- 확장할 디스크의 사이즈는 자동으로 측정됨
- 마침을 누르면 볼륨 확장 완료

- [1] ROS wiki, <https://wiki.ros.org>
- [2] ROS 2 Documentation, <https://docs.ros.org/en/jazzy/index.html>
- [3] 표윤석 박사님 ROS 강의,  
[https://www.youtube.com/watch?v=ot\\_D9N-H4lQ&list=PLRG6WP3c31\\_VlFtFAxSke2NG\\_DumVZPgw](https://www.youtube.com/watch?v=ot_D9N-H4lQ&list=PLRG6WP3c31_VlFtFAxSke2NG_DumVZPgw)
- [4] 오로카, 오픈 로보틱스 커뮤니티, <https://cafe.naver.com/openrt>
- [5] Ubuntu, <https://ubuntu.com>

# QnA

E-mail : [cmin87394@gmail.com](mailto:cmin87394@gmail.com)

Mobile : 010-3357-8739

**Thank you.**

