

# 클라우드 컴퓨팅 입문

## Compute 2

이 자료는 Elixirr의 사전 서면 승인 없이 외부에 배포하기 위해 그 일부를 배포, 인용 또는 복제 할 수 없습니다.

© Copyright Elixirr

Linux

# What is an Operating System?

운영체제 (Operating System, OS)는 컴퓨터 하드웨어와 소프트웨어 자원을 관리하고 컴퓨터 프로그램에서 서비스를 제공하는 기본 소프트웨어입니다. 이는 사용자와 컴퓨터 하드웨어 간의 중개자 역할을 하며, 애플리케이션이 효율적이고 효과적으로 실행되도록 합니다.

Ask ChatGPT

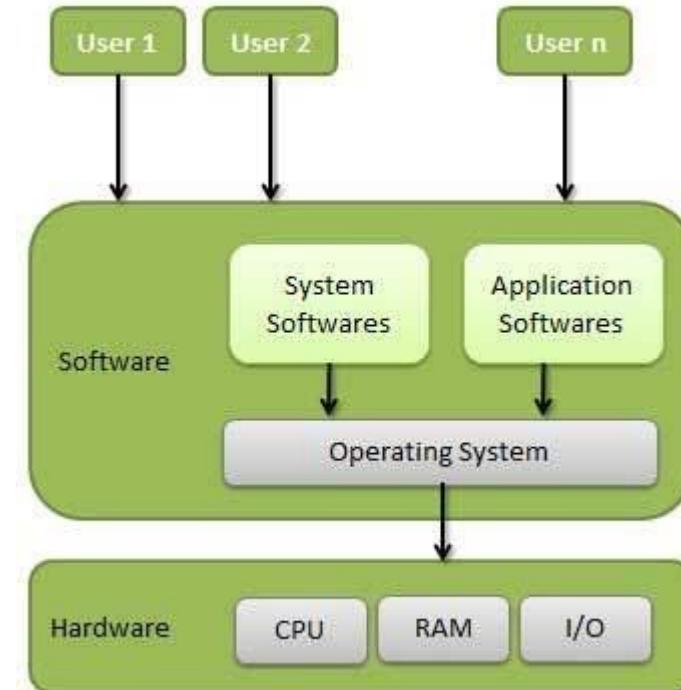
What is an Operating System?

## 운영체제의 핵심 기능

- 프로세스 관리
- 메모리 관리
- 파일 시스템 관리
- 장치 관리
- 보안 및 접근 제어

## 인기 있는 운영체제

- Windows: 사용자 친화적인 인터페이스와 다양한 애플리케이션
- macOS: 디자인과 애플 하드웨어와의 통합으로 잘 알려져 있음
- Linux: 오픈 소스이며, 고도로 커스터마이징이 가능해 서버 및 임베디드 시스템에서 널리 사용
- Android 및 iOS: 모바일 기기 시장에서 지배적이며, 광범위한 애플리케이션 생태계를 제공



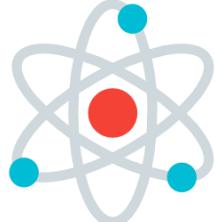
[https://www.tutorialspoint.com/operating\\_system/os\\_overview.htm](https://www.tutorialspoint.com/operating_system/os_overview.htm)

# Linux

리눅스는 오픈소스 기반 운영체제로서 최초에 Linus Torvalds에 의해 개발되었습니다. (Linux는 Linus의 UNIX라는 의미입니다. UNIX는 1970년대에 개발된 대표적인 운영체제입니다.)



Open Source  
(오픈소스)



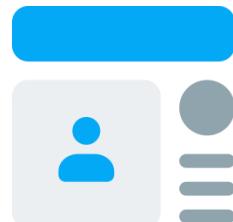
Kernel  
(커널)



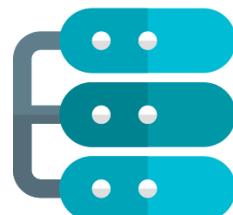
Distributions  
(배포판)



Command Line  
Interface (CLI)  
명령줄 인터페이스



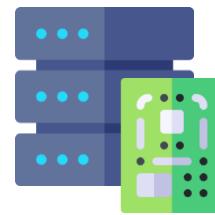
Graphical User  
Interface (GUI)  
(그래픽 사용자 인터페이스)



Multiuser and  
Multitasking  
(다중 유저 및 다중 작업)



Stability and  
Security  
(안정성 및 보안)



Server and  
Embedded  
Systems  
(서버 및 임베디드 시스템)



Compatibility and  
Customizability  
(호환성 및 맞춤 가능성)

# [참고] UNIX

UNIX는 1969년에 AT&T 벨 연구소의 켄 톰프슨(Ken Thompson), 데니스 리치(Dennis Ritchie), 그리고 다른 연구자들에 의해 개발된 운영체제입니다. UNIX는 강력하고 유연한 설계로 인해 현대 컴퓨터 운영체제의 기반을 제공하며, 다양한 변종(예: Linux, macOS, BSD 등)을 통해 널리 사용되고 있습니다.

Ask perplexity

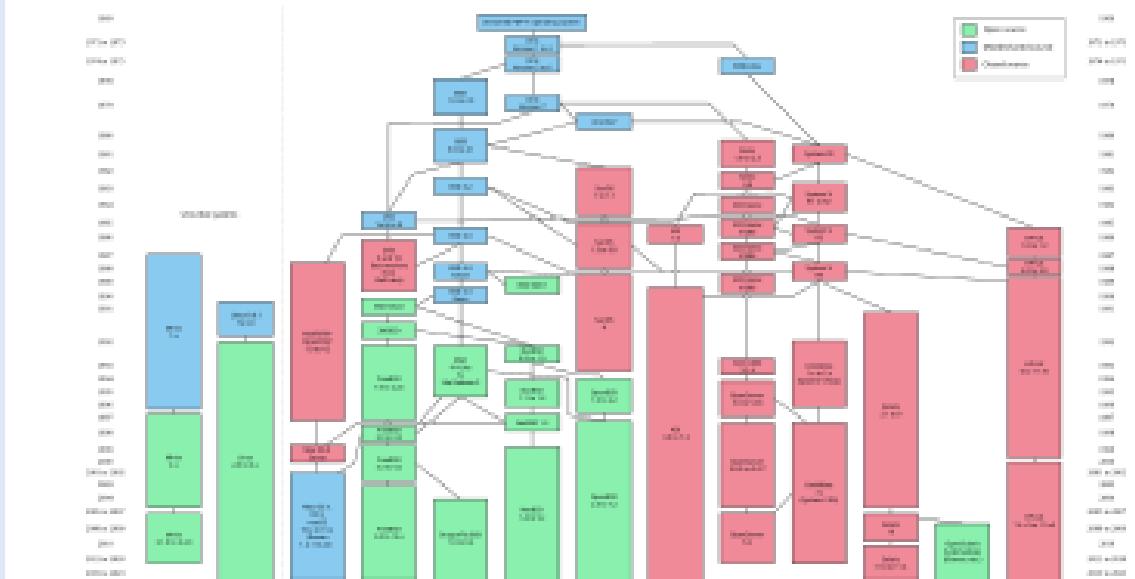
UNIX를 소개해주세요

## 주요 특징

- 멀티유저 시스템: 여러 사용자가 동시에 시스템 자원(CPU, 메모리, 디스크 등)을 사용
- 멀티태스킹 지원: 여러 프로그램을 동시에 실행할 수 있는 기능 제공
- 계층적 파일 시스템: 트리 구조를 기반으로 하는 조직적인 파일 시스템을 지원
- 도구 기반 철학: 각 유ти리티는 단순한 작업을 수행하도록 설계되었으며, 이를 조합하여 복잡한 작업을 처리 가능
- 이식성: 대부분의 UNIX 코드는 C 언어로 작성되어 다른 하드웨어 플랫폼으로 쉽게 포팅할 수 있음

## 주요 용도

- 서버 운영 (웹 서버, 데이터베이스 서버 등)
- 소프트웨어 개발 환경
- 네트워크 장비 운영체제
- 연구 및 교육 목적으로 사용



[https://en.wikipedia.org/wiki/Unix-like#/media/File:Unix\\_history-simple.svg](https://en.wikipedia.org/wiki/Unix-like#/media/File:Unix_history-simple.svg)

# UNIX는? Linux는?

Ask  ChatGPT

UNIX와 Linux 개요 및 유사점 및 차이점을 알려줘

UNIX	LINUX
1960년대와 1970년대에 AT&T의 벨 연구소에서 Ken Thompson, Dennis Ritchie 및 다른 사람들이 개발 다중 사용자, 다중 작업 운영 체제로 설계	1991년 Linus Torvalds가 UNIX에서 영감을 받아 독립적으로 개발 성능, 보안, 유연성에 중점을 둔 무료 오픈 소스 대안으로 설계
AIX(IBM), HP-UX(휴렛팩커드), Solaris(선 마이크로시스템즈)와 같은 많은 버전	Ubuntu, Fedora, Debian, CentOS와 같은 수많은 배포판
초기에는 독점적이었지만, 일부 버전은 오픈 소스로 전환(BSD 등).	GNU GPL 하에 배포되어 누구나 무료로 사용
유사점	차이점
UNIX와 Linux 모두 UNIX계열 운영체제로, 많은 핵심 개념과 설계 원칙 공유	UNIX는 1960년대부터 개발 역사가 시작된 반면, Linux는 1990년대 초
여러 사용자가 동시에 여러 프로세스를 실행	Linux는 Linus Torvalds와 커뮤니티에 의해 처음부터 개발되었고, UNIX는 다양한 가지와 독점 코드베이스를 가진 긴 역사
애플리케이션 수준에서 호환성을 보장하는 POSIX 준수를 목표	UNIX는 전통적으로 독점적이었지만 오픈 소스 버전도 존재하며, Linux는 항상 GPL 하에 오픈 소스로 배포
두 시스템 모두 bash, csh, ksh와 같은 유사한 셸 환경을 사용하여 명령 줄 작업을 수행	Linux 배포판은 사용자 인터페이스, 패키지 관리 및 포함된 소프트웨어 측면에서 다를 수 있는 반면, UNIX 버전은 각 변형 내에서 더 표준화

# [참고] GNU

GNU는 UNIX에 대한 대안으로 자유 소프트웨어 (free software)로 Richard Stallman에 의해서 개발되었습니다. 참고로 GNU는 **GNU's Not Unix**라는 재귀적 약자(recursive acronym)로 해커들 사이에 흔한 표현법입니다.

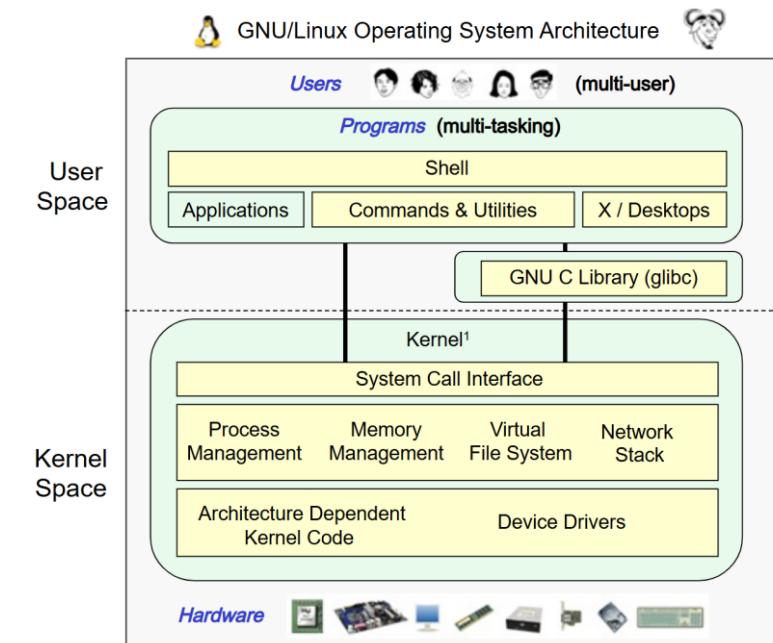


[https://en.wikipedia.org/wiki/Richard\\_Stallman](https://en.wikipedia.org/wiki/Richard_Stallman)



[https://commons.wikimedia.org/w/index.php?title=File:The\\_GNU\\_logo.png&oldid=100000000](https://commons.wikimedia.org/w/index.php?title=File:The_GNU_logo.png&oldid=100000000)

- Richard Stallman은 누구나 무료로 복사하고 수정할 수 있는 완전한 운영체제를 만들고자 함
- 우리가 사용하는 OS는 kernel 부분과 각종 도구들이 포함되는데, GNU에서는 다양한 도구들을 만들었음 (대표적으로 GNU Emacs, GCC)
- Kernel로는 그 당시 유행하던 microkernel이라는 구조를 채용한 GNU HURD를 개발하고 있었지만, 성능이나 구현의 복잡도 등의 문제로 자연되고 있었음
- Linus Torvalds가 만든 Linux Kernel과 GNU의 여러 도구/라이브러리들이 합쳐지면서 본격적인 OS로서의 Linux가 만들어짐
- Richard Stallman이 주장한 free software의 아이디어는 GNU GPL (General Public License)를 통해서 이어지고 있음



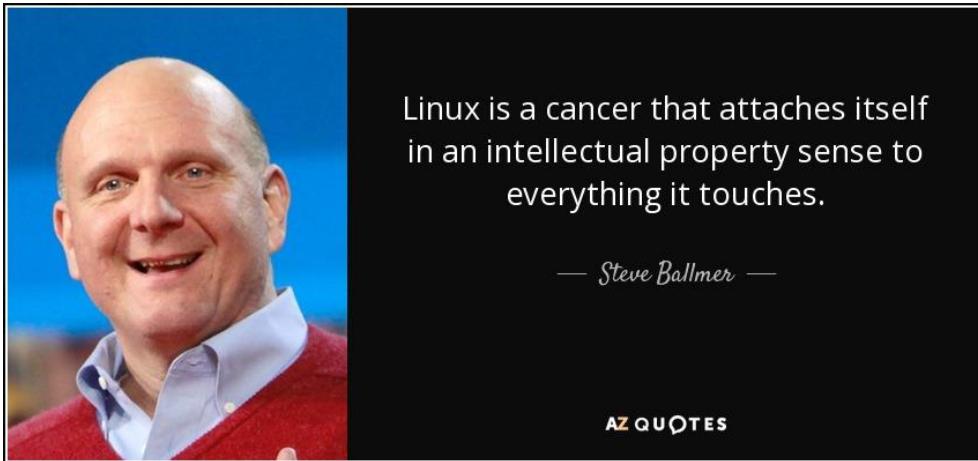
<https://slideplayer.com/slide/7335359/>

# Microsoft and Linux

Microsoft는 Windows라는 고유의 운영체제를 보유한 회사입니다. 반면에 Linux는 Open Source 기반의 운영체제로서 Windows에서 실행되는 앱들이 바로 실행되지 않는 환경입니다. Microsoft와 Linux는 어떤 관계일까요?

2001년 당시 CEO였던 Steve Ballmer는 Chicago Sun-Times와의 인터뷰에서 'Linux는 지적재산권 측면에서 만지는 모든 것에 불는 암입니다'라고 이야기한 바 있습니다.

2014년 당시 (그리고 현재도) CEO였던 Satya Nadella는 San Francisco에서 진행된 마이크로소프트의 클라우드 전략 발표에서 'Microsoft는 Linux를 사랑합니다'라고 이야기했습니다.



<https://www.azquotes.com/quote/665224>



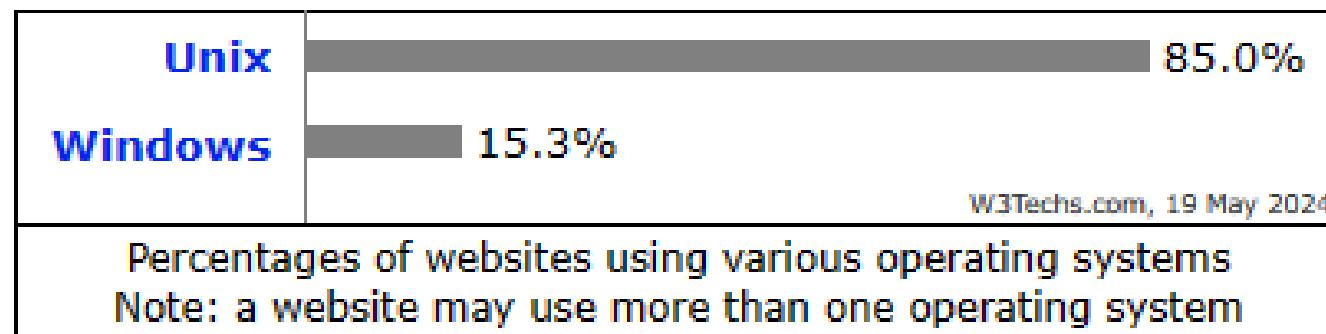
<https://www.theverge.com/2016/9/15/12926288/microsoft-really-does-love-linux>

# PC 환경에서는 Windows, 하지만 서버 환경에서는 UNIX (혹은 UNIX like)

PC 환경에서는 아직도 대부분의 사용자가 Microsoft Windows를 사용하고 있지만, 서버 환경에서는 UNIX 혹은 UNIX와 유사한 운영체제, 특히 Linux가 대세입니다. 그리고 이 추세는 증가하고 있습니다.

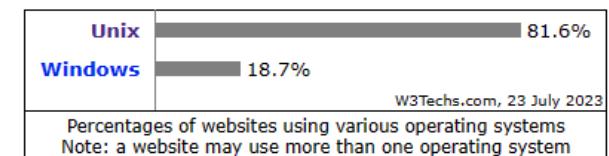
How to read the diagram:

Unix is used by 85% of all the websites whose operating system we know.



\* 여기서 Unix는 Unix like 운영체제를 말함

[https://w3techs.com/technologies/overview/operating\\_system](https://w3techs.com/technologies/overview/operating_system)



# Linux Distros

Linux는 커널과 그 위에 존재하는 사용자 환경이나 어플리케이션이 분리되어 있어서 엄청나게 많은 distro들이 존재하며, 이는 Linux를 처음 접하는 사람에게 너무나 많은 선택지를 제공하면서 초기 접근을 어렵게 하는 이유 중 하나입니다.



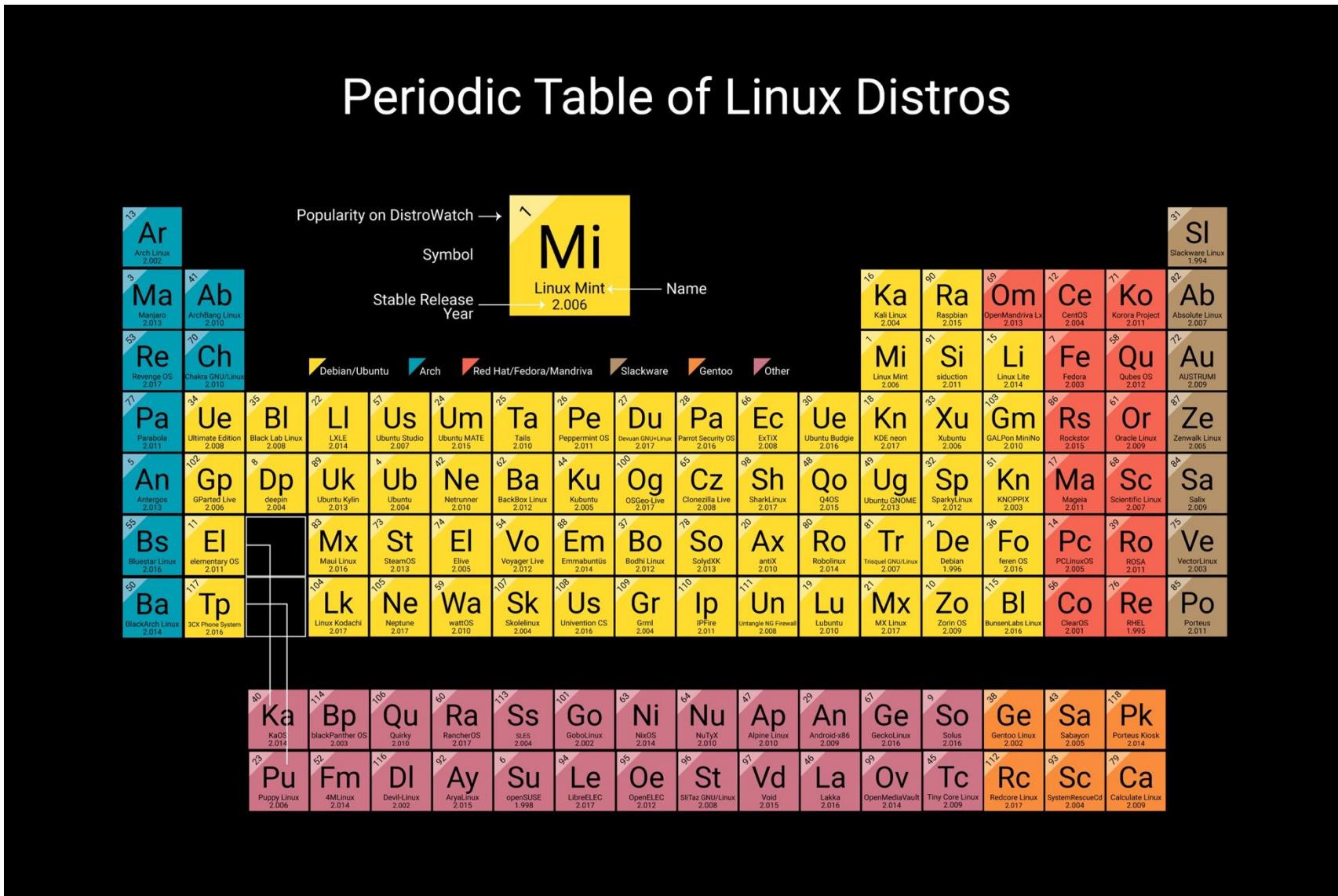
Image Source: <https://medium.com/@kirushikanketheeswaran/linux-distro-1aac3b5937d5>

참고

**List of Linux distributions**

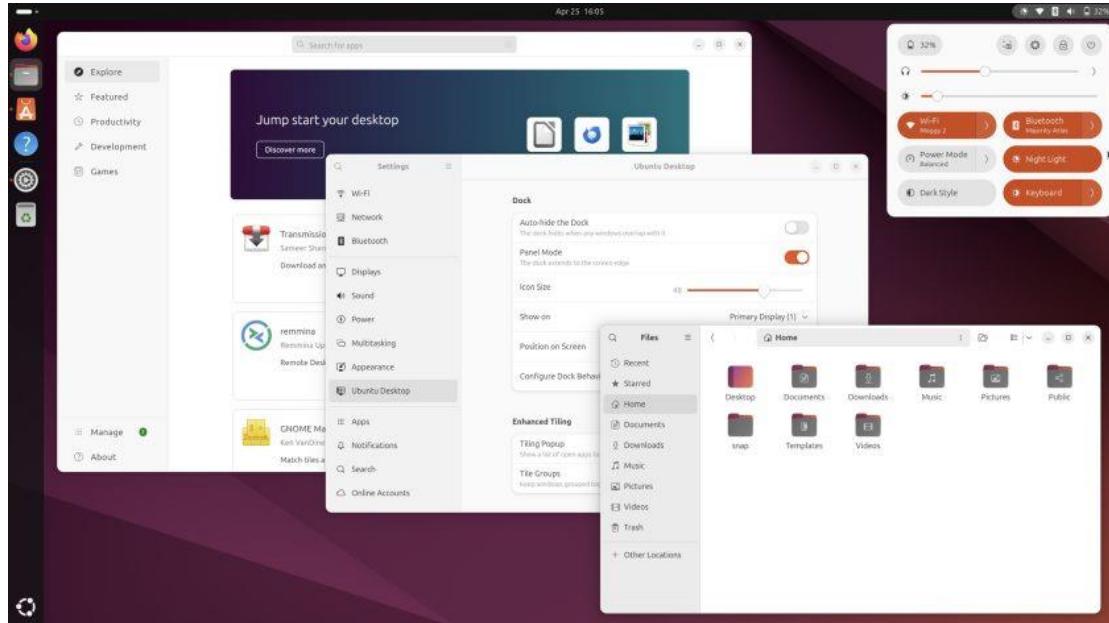
[https://en.wikipedia.org/wiki/List\\_of\\_Linux\\_distributions](https://en.wikipedia.org/wiki/List_of_Linux_distributions)

# 주기율표 형태로 구성한 Linux Distros



# Ubuntu

Ubuntu는 2004년 최초로 릴리즈된 Linux 배포판입니다. Debian 계열로서 Debian과 많은 유사점이 있습니다.



<https://www.omgubuntu.co.uk/2024/04/ubuntu-24-04-released>

## Ubuntu 22.04 LTS

Ubuntu 21.10

Ubuntu 21.04

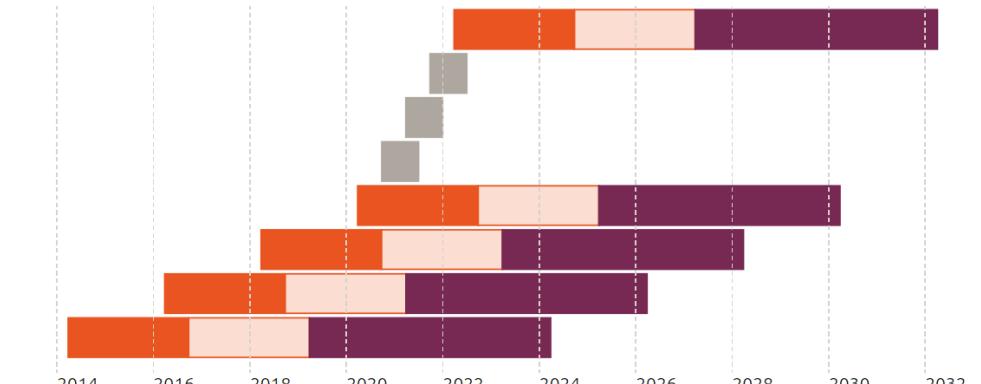
Ubuntu 20.10

**Ubuntu 20.04 LTS**

**Ubuntu 18.04 LTS**

**Ubuntu 16.04 LTS**

**Ubuntu 14.04 LTS**



Orange: Hardware and maintenance updates

Light Orange: Maintenance updates

Grey: Interim release Standard Support

Dark Purple: Extended Security Maintenance (ESM)

<https://ubuntu.com/blog/what-is-an-ubuntu-lts-release>

안정적인 운영을 위해서는 LTS (Long Term Support) 버전을 사용하는 것이 좋습니다. 예를 들어 22.04 LTS는 2022년 4월에 출시된 버전으로 오랜 기간 지원될 버전입니다.

# Linux 핵심 요소

# Linux의 주요 구성 요소 - Kernel

Linux 커널은 컴퓨터 하드웨어와 소프트웨어 사이의 인터페이스 역할을 하는 Linux 운영 체제의 핵심 구성 요소입니다. 1991년 Linus Torvalds가 최초로 오픈소스로 개발한 이후 전 세계 개발자 커뮤니티에서 지속적으로 개선되어 왔고, 클라우드 컴퓨팅에서 사용되는 대표적인 운영체제입니다.



What is Linux Kernel?

## 주요 특징

Linux 커널은 몇 가지 중요한 작업을 수행합니다

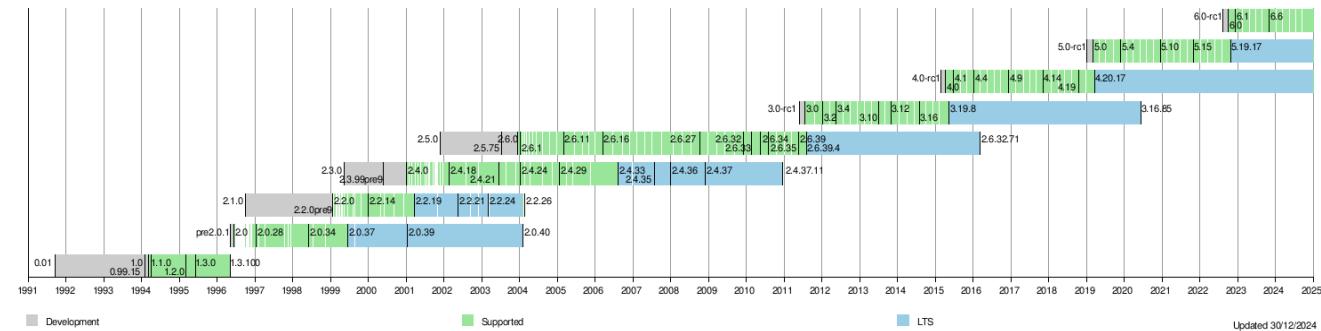
- 메모리 관리: 메모리 사용량 추적 및 할당
- 프로세스 관리: 다양한 프로세스에 대한 CPU 사용량 결정
- 장치 드라이버: 하드웨어와 프로세스 간 중개
- 시스템 호출 및 보안: 프로세스의 서비스 요청 처리

## 아키텍처

- Linux 커널은 Monolithic 커널로, 단일 주소 공간에서 작동
- 유연하게 설계되어 특정 요구 사항이나 프로그램 요구 사항에 맞게 수정 가능

## 버전

- 1991년 9월 17일 버전 0.01을 시작으로 1994년 3월에 버전 1.0이 출시되었고, 현재는 6.12까지 나온 상태



[https://en.wikipedia.org/wiki/Linux\\_kernel\\_version\\_history](https://en.wikipedia.org/wiki/Linux_kernel_version_history)

## Linux의 주요 구성 요소 - Shell

Linux 셸은 **사용자와 운영 체제의 커널 사이의 중개자** 역할을 하는 강력한 명령줄 인터페이스입니다. 셸은 사람이 읽을 수 있는 명령을 해석하여 커널이 이해하고 실행할 수 있는 명령으로 변환합니다.



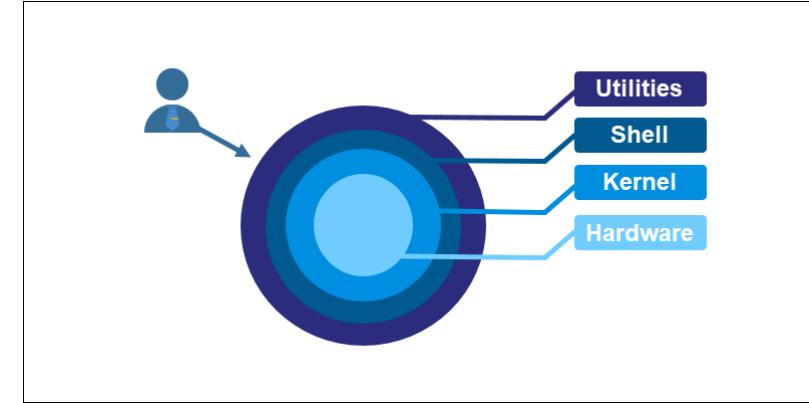
# What is Linux Shell?

## 주요 특징 및 기능

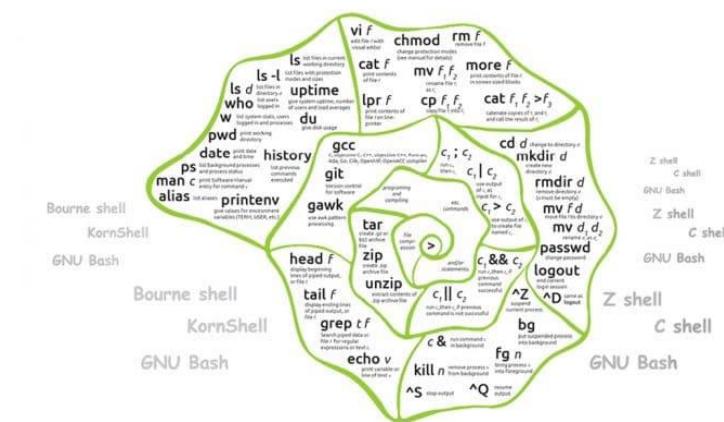
- 명령 해석: 셸은 명령의 형태로 사용자 입력을 받아 해석한 후 실행을 위해 운영 체제로 전송
- 스크립팅 기능: 셸은 단순한 명령 실행을 넘어 스크립팅을 지원함으로 사용자가 복잡한 프로그램을 만들어 작업을 자동화하고 고급 작업 수행 가능
- 기본 제공 명령: Linux 셸에는 파일 조작, 텍스트 처리 및 시스템 관리를 위한 다양한 기본 제공 명령이 있음
- 입력/출력 리디렉션: 셸은 입력 및 출력을 리디렉션하는 기능을 제공하여 사용자가 명령을 연결하고 데이터 흐름 조작 가능
- 환경 사용자 지정: 사용자는 변수, 별칭 및 함수를 사용하여 셸 환경을 사용자 지정하여 자신의 환경에 맞게 조정 가능

## Shell 종류

- Bourne Shell, Bash (Bourne Again Shell), Z Shell (zsh), KornShell (ksh) 등



<https://phoenixnap.com/kb/linux-shells>



<https://www.opensourceforu.com/2020/08/a-list-of-linux-shells-to-choose-from/>

# Linux의 주요 구성 요소 - File system

Linux 파일 시스템은 **저장 장치에 데이터를 저장, 구성 및 액세스하는 방법**을 관리하는 Linux 운영 체제의 기본 구성 요소입니다. "/"(루트) 디렉터리에 뿌리를 두고 다른 모든 파일과 디렉터리가 분기되는 통합 계층 트리 구조로 되어 있습니다.



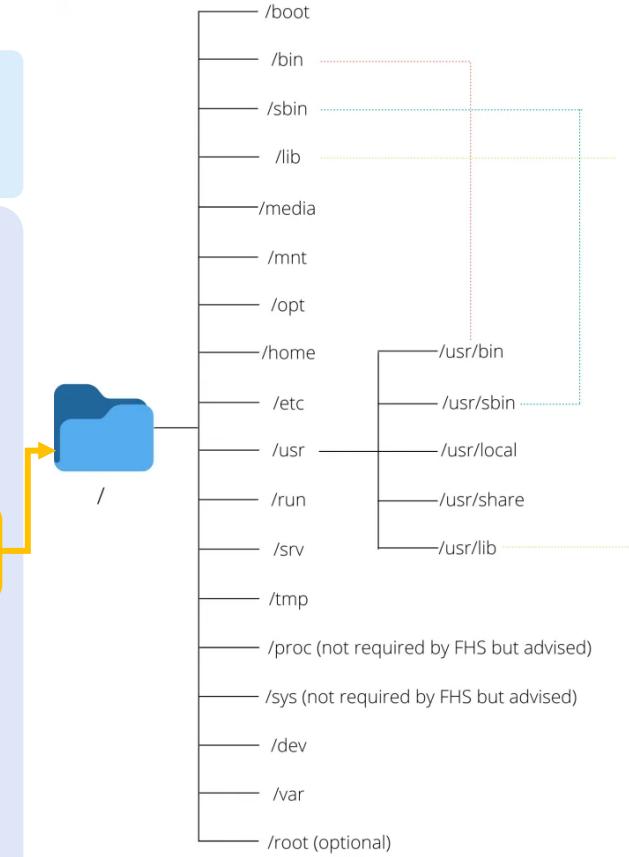
What is Linux File system?

## 주요 특징

- 통합 구조: 마운트된 모든 파일시스템은 루트 디렉토리(/)를 최상위 디렉토리로 하여 단일 트리에 통합
- 모든 것이 파일: Linux는 디렉토리, 하드웨어 장치, 심지어 활성 프로세스까지 포함한 모든 것을 파일로 취급
- 파일시스템 계층 표준(FHS): Linux 배포판 전반의 디렉토리 구조와 내용을 정의하여 일관성과 호환성을 보장

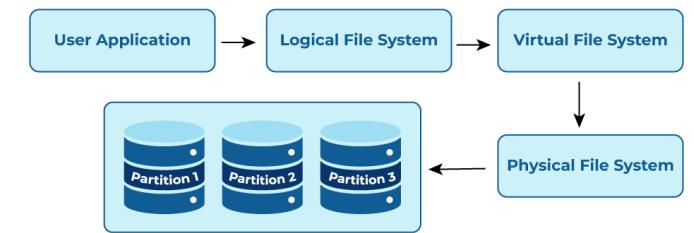
## 구성 요소

- 논리 파일 시스템: 사용자 애플리케이션과 파일 시스템 간의 인터페이스 역할을 하며 파일 열기, 읽기, 닫기 등의 작업을 관리
- 가상 파일 시스템(VFS): 호환성을 위한 표준화된 인터페이스를 제공하여 여러 개의 물리적 파일 시스템을 동시에 작동할 수 있도록 함
- 물리적 파일 시스템: 디스크의 물리적 메모리 블록을 관리 및 저장하고, 저수준 데이터 저장 및 검색을 처리



<https://microsoft.github.io/WhatTheHack/020-LinuxFundamentals/Student/resources/fhs.html>

## The Architecture of a File System



<https://www.geeksforgeeks.org/linux-file-system/>

# 경로 (Path)

Linux에서 경로는 파일 시스템 계층 구조 내에서 파일 또는 디렉터리의 위치를 지정하는 문자열입니다. Linux 시스템에서 파일을 탐색하고 관리하려면 경로를 이해하는 것이 중요합니다.

**Ask perplexity**

### What is Linux Path?

**유형**

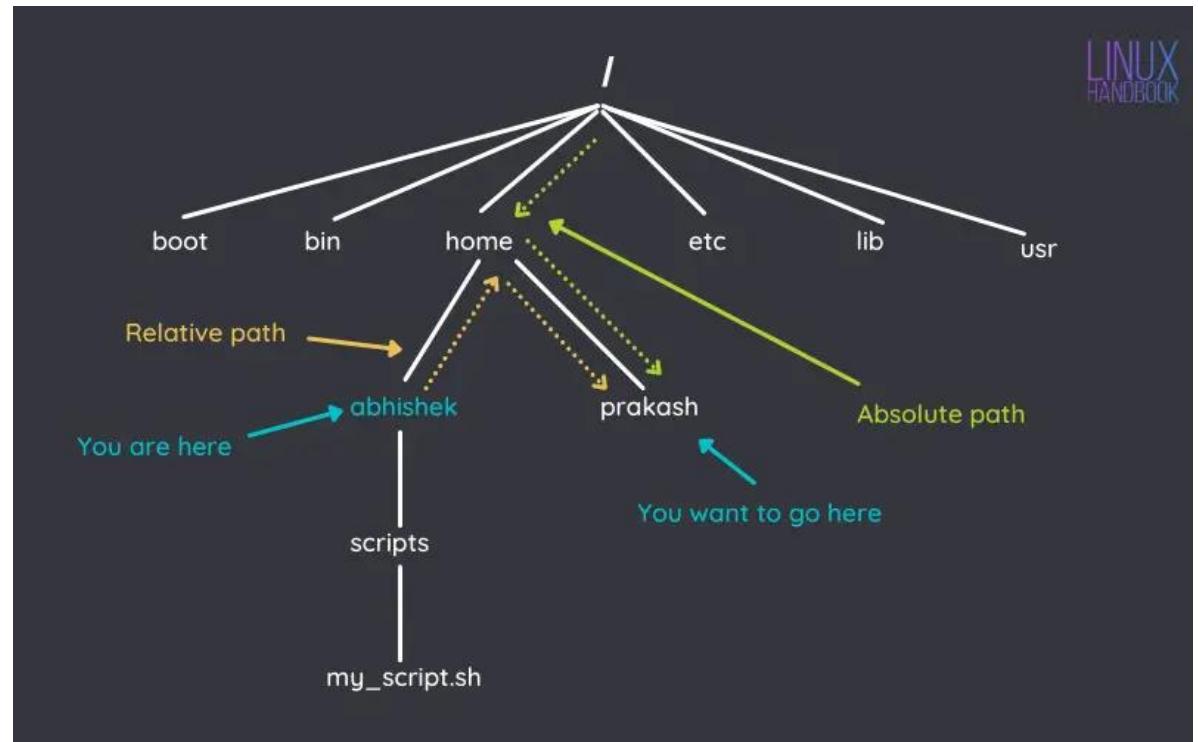
- 절대 경로:** 이는 루트 디렉터리(/)에서 시작하여 파일 또는 디렉터리의 정확한 위치를 지정 예) "/home/user/documents/file.txt"
- 상대 경로:** 현재 작업 디렉터리를 기준으로 파일 또는 디렉터리의 위치를 지정 예) "documents/file.txt"

**경로 구성 요소**

- 경로는 **슬래시(/)로 구분된 디렉토리 이름**으로 구성
- 경로의 각 디렉터리 이름 → "경로 이름 구성 요소"  
예) "/path/to/file/" 경로에서 구성 요소는 "path", "to", "file"

**PATH 변수**

- PATH 환경 변수는 명령을 실행할 때 시스템이 검사하는 디렉터리 목록이 포함됨
- 사용자가 명령을 입력하면 Linux는 해당 실행 파일을 찾기 위해 이 디렉터리를 검색함
- echo \$PATH** 명령을 사용하여 현재 경로를 확인 가능



<https://linuxhandbook.com/absolute-vs-relative-path/>

\* **home** directory: 사용자들에게 할당된 디렉토리 (Windows의 users에 해당)

# Linux File Permission

Linux 파일 액세스 권한은 운영 체제 보안 모델의 중요한 측면입니다. 파일과 디렉터리를 읽고, 쓰고, 실행할 수 있는 사용자를 제어하여 시스템 무결성과 데이터 보호를 보장합니다.



## Linux File Permission

### 기본 권한

- 읽기(r): 파일의 내용을 보거나 디렉터리 내용을 나열
- 쓰기(w): 파일을 수정 또는 삭제하거나 디렉토리 내에서 파일을 추가/제거
- 실행(x): 파일을 프로그램 또는 스크립트로 실행하거나 디렉터리에 액세스

### 권한 카테고리

- 소유자: 파일 또는 디렉터리를 만든 사용자
- 그룹: 공유 액세스 권한을 가진 사용자 집합
- 기타: 시스템의 다른 모든 사용자

### 권한 보기

파일 권한을 보려면 터미널에서 `ls -l` 명령을 사용

-rwxrw-r—

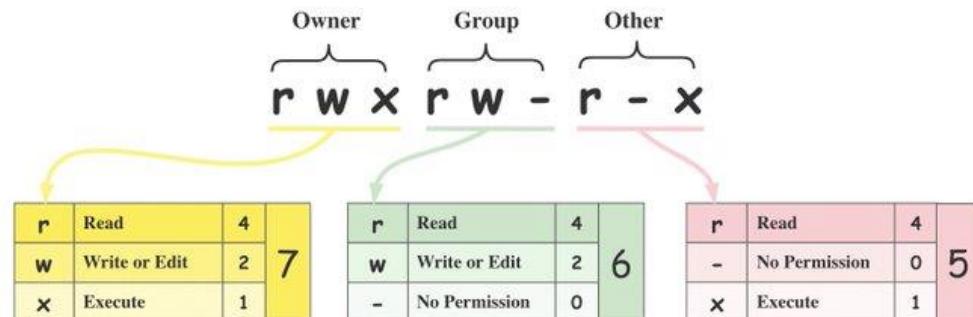
### 권한 변경 명령

`chown` – change owner / `chgrp` – change group

## Linux File Permissions

blog.bytebytego.com

Binary	Octal	String Representation	Permissions
000	0 (0+0+0)	---	No Permission
001	1 (0+0+1)	--x	Execute
010	2 (0+2+0)	-w-	Write
011	3 (0+2+1)	-wx	Write + Execute
100	4 (4+0+0)	r--	Read
101	5 (4+0+1)	r-x	Read + Execute
110	6 (4+2+0)	rw-	Read + Write
111	7 (4+2+1)	rwx	Read + Write + Execute



<https://x.com/alexubyte/status/1532026886359879681?mx=2>

# 명령어 실행 기본

명령어 입력 기본 구문은 다음의 세가지 유형이 있습니다.

## 1. 명령

- 명령만을 입력. 추가적인 옵션이 없어서 복잡한 작업은 수행할 수 없지만 사용하기 쉬움

예) Is

## 2. 명령 + 인수

- 대상이 되는 파일명, 디렉토리명 등의 문자열 (인수)를 지정
  - 명령과 인수 사이에는 **스페이스**로 구분

예) ls /home

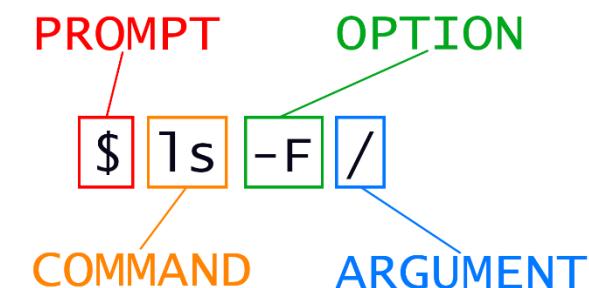
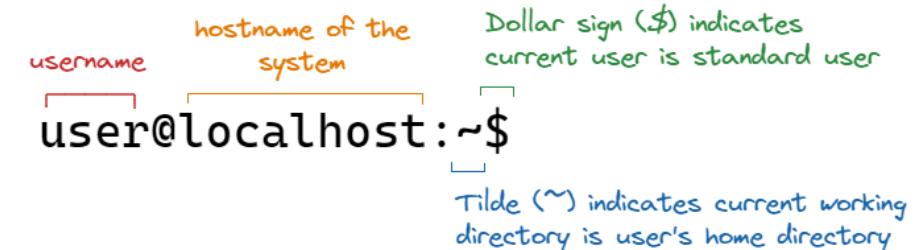
### 3. 명령 + 옵션 + 인수

- 명령에 옵션을 붙여 기능을 확대. 통상적으로 하이픈(-) 기호가 사용됨
  - 명령 옵션 인수 사이에는 모두 스페이스로 구분

예) ls -a /home

TIP

- 디렉토리명, 파일명 등을 입력할 때 **TAB** 키를 눌러서 자동완성 기능을 사용해보기!



<https://github.com/ssbcb/Linux-for-biologists-workshop/blob/main/Introduction-to-linux-101/Introduction-to-linux-101.md>

# ASCII, Unicode

ASCII(American Standard Code for Information Interchange)와 Unicode는 컴퓨터에서 텍스트를 표현하는 데 사용되는 문자 인코딩 표준입니다. 다만, ASCII에서 표현할 수 있는 문자가 제한적이므로 이를 극복하기 위해서 Unicode가 나오게 되었습니다.



ASCII? Unicode?

## ASCII

- 1963년에 개발된 ASCII는 7~8비트를 사용하여 최대 256 자를 표현하는 간단한 인코딩 체계로, 주로 영문자, 숫자, 기본 문장 부호에 중점을 두어 표현
- 한글과 같은 복잡한 문자를 표현할 공간이 부족 → 현대 한글의 가능한 조합은 11,172자

## Unicode

- 유니코드는 여러 언어와 스크립트에서 약 155,000개의 문자를 지원하는 보다 포괄적인 문자 인코딩 표준
- 특수 기호, 이모티콘, 다양한 문자 체계의 문자 등 거의 모든 주요 언어의 텍스트를 표현할 수 있는 보편적인 방법을 제공함으로써 ASCII의 한계를 극복
- UTF-8, UTF-16, UTF-32 등의 다양한 인코딩 방법 제공



강	깂	겼	것	겐	겟	결	경	결	곰	곳
AC53	AC50	ACT0	AC51	AC52	AC54	AC55	AC56	AC57	AC58	AC59
개	캠	건	겄	겔	켁	결	계	겜	꼰	꼰
AC54	AC56	ACT4	AC55	AC57	AC58	AC59	AC50	AC51	AC52	AC53
객	갭	겁	겅	겝	겔	결	켁	겜	꼰	꼰
AC55	AC56	ACT5	AC58	AC59	AC54	AC55	AC56	AC57	AC58	AC59
객	꽱	꽱	꽱	꽱	꽱	꽱	꽱	꽱	꽱	꽱
AC56	AC58	ACT6	AC59	AC58	AC55	AC56	AC57	AC58	AC59	AC59
갯	갯	갯	갯	갯	갯	갯	갯	갯	갯	갯
AC57	AC57	ACT7	AC57							
갠	갠	갠	겻	꿰	겡	결	결	겜	꼰	꼰
AC58	AC58	ACT8	AC58							

<https://www.techmonitor.ai/what-is/what-is-unicode-4942062>

## [TIP] 파워포인트에서 긁어서 붙여 넣기를 했는데 오류가 나는 이유

파워포인트에서는 Smart Quotes라는 기능이 있어서, 사용자가 입력한 따옴표, 하이픈기호 등을 문맥에 맞게 자동으로 바꿔주는데, 이 기능이 활성화되면 Linux Command를 복사하는 경우 원하지 않는 코드가 들어가게 됩니다.

Powerpoint에  
의해 자동  
변환  
되는 경우

```
ls -a → 006C 0073 0020 002D 0061  
ls -a → 006C 0073 0020 2013 0061
```

1. 직접 타이핑을 하거나
2. Ctrl + Shift + V로 붙여 넣거나
3. 옵션 -> 언어 교정 -> 자동고침 옵션 기능을 Turn Off

Paste text only.

Ctrl+Shift+V

<https://support.microsoft.com/en-us/office/keyboard-shortcuts-in-word-95ef89dd-7142-4b50-afb2-f762f663ceb2>

Dash Punctuation 카테고리			
Character List			
U+002D	U+058A	U+05BE	U-
-	-	-	
Hyphen-Minus	Armenian Hyphen	Hebrew Punctuation Maqaf	Canadi
U+2011	U+2012	U+2013	U-
-	-	-	
Non-Breaking Hyphen	Figure Dash	En Dash	Ei
U+2E1A	U+2E3A	U+2E3B	U+
--	--	--	

<https://www.compart.com/en/unicode/category/Pd>

# [참고] long dashes, short dashes, hyphens

## 하이픈(hyphen), 엔 대쉬(en dash), 엠 대쉬(em dash) 사용법

대쉬 중 가장 짧고, 또 가장 많이 쓰이는 것이 하이픈(-, Hyphen)입니다.

하이픈은 두 개 이상의 단어를 하나로 묶어주는 기능을 합니다. 예를 들어, 'a test-tube', '4-aminobenzensulphonic acid', 'fail-safe', 'state-of-the-art facilities' 같은 단어들이 그 예입니다.

또 '15-inch rod', '2-month vacation' 과 같이 값(숫자)을 명사와 연결해 한 단어로 만들 때 하이픈(Hyphen)이 쓰입니다. (이때 SI 단위는 제외됩니다.) 하이픈 하나로 연결된 단어는 대부분의 경우 하이픈 앞에 들어가는 단어가 뒤의 단어를 수식한다는 점을 기억해 두십시오.

중간 길이를 가진 대쉬는 엔 대쉬(–, En dash)로, '6–10 kg', '6–10 mm' 처럼 범위를 나타내는 데 쓰입니다.

참고문헌을 작성할 때 *Information Design Journal 17: 153–154*와 같이 엔 대쉬(En Dash)를 사용해 페이지 수를 나타낼 수 있습니다. 이런 경우에 엔 대쉬는 전치사 *to*를 대체한다고 볼 수 있습니다.

엠 대쉬(Em Dash)의 또 다른 기능은 '미켈리스–멘텐 방정식 Michaelis–Menten equation', '에너지–환경 인터페이스 energy–environment interface', '유전·환경 논쟁 nature–nurture debate'처럼 중요도가 똑 같은 두 개의 단어를 결합시키는 데 쓰입니다. 즉, 엠 대쉬를 사용할 때는 앞의 단어와 뒤의 단어 사이에 수식관계가 없습니다.

마지막, 제일 긴 줄표가 엠 대쉬(—, Em dash)로, 과학 논문에서는 사용되는 경우가 드뭅니다.

목록이나 설명 앞에 나오는 콜론(:)과 대응관계라고 볼 수 있습니다. 엠 대쉬는 이런 항목들 다음에 나오는 부호로, 앞에 나열한 항목들을 요약하거나 대조되는 사실을 끌어냅니다. 예를 들면 다음과 같습니다.

예)

"태양열 히터, 풍력 터빈, 수차, 심지어 마른 가지를 모아 태워 불인 장작불에 이르기까지 공통의 특성을 가지고 있다—모두 재생가능한 에너지원을 사용한다는 것이다."

Solar water heaters, wind turbines, water mills, or even a simple fire made by burning a few dried branches, all share a common feature—they all use renewable sources of energy."

[https://blog.naver.com/editage\\_kr/70185267400](https://blog.naver.com/editage_kr/70185267400)

- en dash – 'n'의 폭과 같은 dash
- em dash – 'm'의 폭과 같은 dash



Microsoft 365 Life Hacks > Writing > The Types of Dashes—and How to Use Them

August 06, 2021

## The Types of Dashes—and How to Use Them

Between long dashes, short dashes, and hyphens—and all of their various uses—it can be difficult to keep the rules straight and stick to correct [grammar](#) in your writing.



In this guide to the types of dashes, we cover each of them—em dashes, en dashes, and hyphens—and offer examples that you can apply in your own work to communicate your best ideas. Whether you're [writing a cover letter](#) or the next great novel, you'll soon be an expert on how and when to use dashes.



Microsoft 365  
Write with Confidence using Editor

Get started with Microsoft 365  
It's the Office you know, plus the tools to help you work better together, so you can get more done—anytime, anywhere.  
[Buy Now >](#)

<https://www.microsoft.com/en-us/microsoft-365-life-hacks/writing/types-of-dashes>

# redirection

Linux 리디렉션은 사용자가 명령의 입력과 출력을 제어하여 명령줄 인터페이스의 유연성과 기능을 향상시킬 수 있는 강력한 기능입니다. 이 기능을 사용하면 **명령의 표준 입력, 출력 및 오류 스트림을 변경하여 파일이나 다른 명령으로 리디렉션할 수 있습니다.**



## Linux redirection

### 표준 스트림

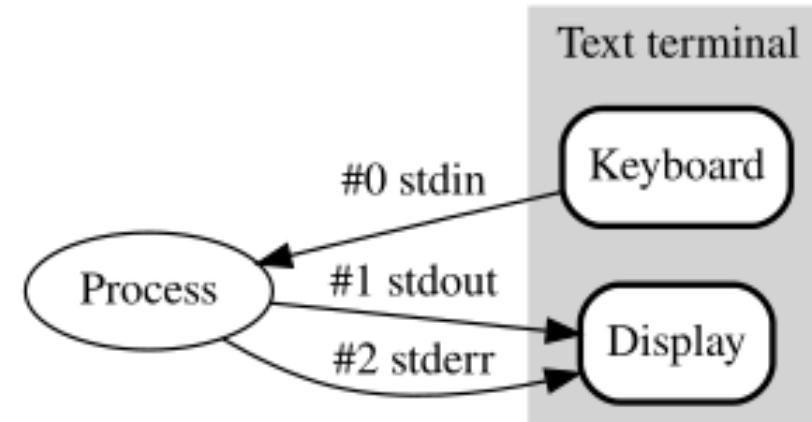
- 표준 입력(stdin): 보통 키보드
- 표준 출력(stdout): 일반적으로 터미널 화면
- 표준 오류(stderr): 기본적으로 역시 터미널 화면

### 리디렉션 오퍼레이터

- > 표준 출력을 리디렉션하여 대상 파일을 덮어쓰기
- >> 표준 출력을 리디렉션하여 대상 파일에 추가하기
- < 파일에서 표준 입력을 리디렉션하기

### 사용 예

- |                         |                       |
|-------------------------|-----------------------|
| • ls -al > dummy.txt    | dummy.txt 결과 확인해보기    |
| • ls -al >> dummy.txt   | dummy.txt 결과 다시 확인해보기 |
| • head -n 5 < dummy.txt | 출력 결과 확인해보기           |



[https://en.wikipedia.org/wiki/Redirection\\_%28computing%29](https://en.wikipedia.org/wiki/Redirection_%28computing%29)

# pipe

Linux 파이프는 사용자가 여러 명령을 서로 연결하여 한 명령의 출력을 다른 명령의 입력으로 리디렉션할 수 있는 강력한 기능입니다. 이를 통해 복잡한 데이터 처리 워크플로우를 생성하고 효율적인 명령줄 작업을 수행할 수 있습니다.



## Linux pipe

### 기본 개념

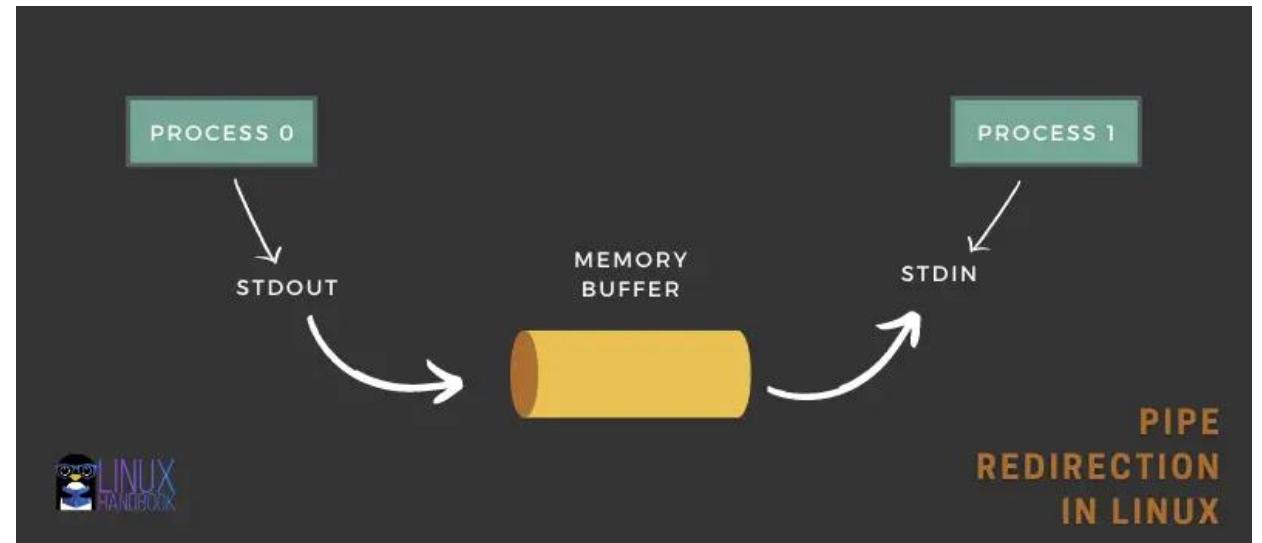
- Linux에서 파이프는 세로 막대 기호 “|”로 표시되며, 한 명령의 표준 출력(stdout)을 다른 명령의 표준 입력(stdin)에 연결하는 데 사용
- 이렇게 하면 명령의 '파이프라인'이 생성되어 데이터가 시퀀스의 각 명령을 통해 왼쪽에서 오른쪽으로 흐름

### 기본 문법

- Command-1 | Command-2 | ... | Command-N

### 파이프 종류

- Anonymous pipe : 위에서 설명한 내용으로 임시로 생성되는 파이프
- Named pipe : mkfifo라는 명령으로 생성되는 파이프로 삭제하기 전까지 지속되는 파이프



<https://linuxhandbook.com/pipe-redirection/>

# Super user

Windows의 Admin에 해당하는 것이 Linux의 superuser입니다. su (Substitute User – 최초에는 SuperUser였음)나 sudo (SuperUser DO 또는 Substitute User DO) 등의 명령어를 통해서 superuser 권한으로 실행할 수 있습니다.

**Ask perplexity**

## Linux su / sudo

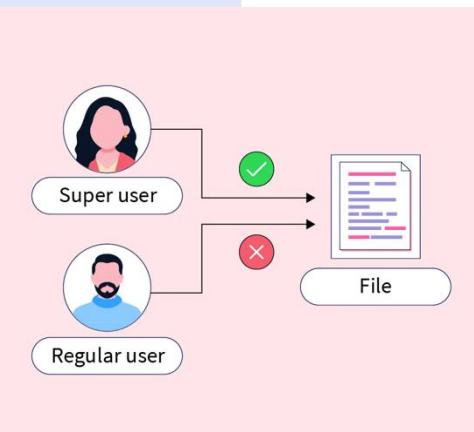
**su**

- 인수를 지정하지 않고 실행하면 superuser로 switch
- 전환하고자 하는 계정의 비밀번호를 입력해서 전환
- 환경 변수 등을 포함한 전체 사용자 컨텍스트를 전환

**sudo**

- 현재 계정의 비밀번호를 입력
- 단, 현재 계정이 sudo group에 있어야 함
- 명령어 별로 권한 지정 가능

<https://www.scaler.com/topics/super-user-in-linux/>



Parameters	sudo	su
Type	Privilege authorization	Command
Operating System	Unix-like	Unix and Unix-like
Function	To seek root permission through a single command	To seek root permission through switching to superuser or root user
Developer	Todd C. Miller	AT&T Bell Laboratories
Author	Robert Coggeshall and Cliff Spencer	Dennis Ritchie and Ken Thompson

<https://kr.pinterest.com/pin/difference-between-sudo-and-su-commands-in-linux--364650901093506093/>

# 리눅스 기본 명령어

Linux 기본 명령어를 정리해 놓은 사이트는 '매우' 많고, 책도 많이 있습니다. Linux command는 자주 바뀌지 않기 때문에, 좀 오래된 자료를 찾아봐도 무방하며, 본인에게 편한 사이트나 책을 찾아서 보면 됩니다.

명령어	설명
cd	Change Directory. 현재 디렉토리의 위치를 변경
pwd	Print Work Directory. 현재 디렉토리의 절대 경로를 표시
ls	LiSt directory. 디렉토리의 정보를 조사
mv	MoVe file. 파일명을 변경하거나 이동
cp	CoPy file. 파일을 복사
mkdir	MaKe DIRectory. 새로운 디렉토리를 생성
rmdir	ReMove DIRectory. 빈 디렉토리를 삭제
rm	ReMove file. 파일이나 디렉토리를 삭제. '-r'과 같이 사용하는 경우 많음
cat	conCATenate. 파일 내용 열람. '>'(redirection)과 같이 사용하는 경우 많음
grep	Global Regular Expression Print. 정규표현식을 통한 문자열 검색 도구
find	FIND. 파일 및 디렉토리 검색 도구

# Linux 기본 명령어

Linux는 Desktop 사용자를 위한 GUI도 제공 하지만, 대부분 서버 환경에서 사용되며, CLI 환경에서 필요한 핵심 명령어들을 숙지하는 것이 좋습니다.

Basic Commands		File Permissions		Environment Variables		User Management	
<code>ls</code>	List all files in the present working directory	<code>ls -l</code>	Show file type and access permission	<code>echo \$VARIABLE</code>	Display the value of a variable	<code>sudo adduser username</code>	Add a new user
<code>ls -R</code>	Lists files in sub-directories	<code>r</code>	Read	<code>env</code>	Display all environment variables	<code>sudo passwd -l username</code>	Change the password of a user
<code>ls -a</code>	Lists hidden files	<code>w</code>	Write	<code>VARIABLE_NAME=variable_value</code>	Create a new variable	<code>sudo userdel -r username</code>	Remove a newly created user
<code>ls -al</code>	Lists files and directories with detailed information	<code>x</code>	Execute	<code>unset</code>	Remove a variable	<code>sudo usermod -a -G GROUPNAME username</code>	Add a user to a group
<code>cd or cd ~</code>	Go back to home directory	<code>-=</code>	No permission	<code>export Variable=value</code>	Set the value of an environment variable	<code>sudo deluser USER GROUPNAME</code>	Remove a user from a group
<code>cd ..</code>	Move one directory level up	<code>chown user</code>	Change the ownership of a file/directory			<code>finger</code>	Shows information of all the logged in users
<code>cd</code>	Change to a particular directory	<code>chown user:group filename</code>	Change the user as well as group for a file or directory			<code>finger username</code>	Returns information on a specified user
<code>cd /</code>	Go to the root directory						
<code>touch &lt;filename&gt;</code>	Create a new file						
<code>cat filename</code>	Display the file content						
<code>mv filename new_file_name</code>	Renames the file to a new filename						
<code>rm filename</code>	Deletes a file						
<code>sudo</code>	Allow a user to run programs with the security privileges of the admin						
Scripting		VI Editing		Networking		Processes	
<code>awk, gawk</code>	Pattern scanning	<code>i</code>	Insert at cursor	<code>ssh username@ip-address or hostname</code>	Login into a remote Linux machine	<code>bg</code>	Send a process to the background
<code>tsh</code>	Tiny shell	<code>a</code>	Write after cursor	<code>ping hostname="" or ""</code>	Ping and analyze network and host connections	<code>fg</code>	Run a stopped process in the foreground
<code>""</code>	Anything within double quotes is unchanged except \$ and \	<code>A</code>	Write at the end of line	<code>dir</code>	Display files in the current directory of a remote computer	<code>top</code>	Detail all active processes
<code>' '</code>	Anything within single quotes is unchanged	<code>ESC</code>	Terminate insert mode	<code>cd "dirname"</code>	Change directory to "dirname" on a remote computer	<code>ps</code>	Return the status of processes running
<code>source [file]</code>	Load any functions file into the current shell	<code>u</code>	Undo last change	<code>put file</code>	Upload 'file' from local to a remote computer	<code>ps PID</code>	Return the status of a particular process
		<code>x</code>	Delete character at the cursor	<code>get file</code>	Download 'file' from remote to a local computer	<code>pidof</code>	Return the Process ID of a process
		<code>dd</code>	Delete line	<code>quit</code>	Logout	<code>kill PID</code>	Kills a process
						<code>nice</code>	Starts a process with a given priority
						<code>renice</code>	Changes priority of an already running process
						<code>df</code>	Return free hard disk space on your system
						<code>killall</code>	Kill processes by name
						<code>ps aux</code>	Display an overview of processes running
						<code>df -h</code>	Display the file system disk space statistics
File Compression		Misc					
<code>tar xvzf</code>	Create or extract .tar or .tgz files	<code>pwd -P</code>	Print current working directory				
<code>gzip, gunzip, zcat</code>	Create or extract .gz files	<code>bc</code>	High precision calculator				
<code>uuencode, uudecode</code>	Create or extract .Z files	<code>expr</code>	Evaluate expression				
<code>zip, unzip -v</code>	Create or extract .ZIP files	<code>cal</code>	Print Calendar				
<code>rpm</code>	Create or extract .rpm files	<code>export</code>	Assign or remove environment variable				
<code>rar</code>	Create or extract .rar files	<code>' [command] '</code>	Back quote, execute command				
		<code>\$(variable)</code>	If set, access the variable				

<https://images.ctfassets.net/aw6mkmszlj4x/2oDcVTRgDK6KIVBshsKuO8/54d205b6c365cd181430565f3706be9c/linuxthree.png>