

Linux

WeGo & WeCAR

목 차

1. OS (운영 체제)
2. Linux
3. Ubuntu
4. Linux Filesystem
5. Linux Command
6. Etc

01

Operating
System

OS
(운영 체제)

01 OS (운영 체제)

- OS : Operating System
- 사용자가 컴퓨터를 쉽게 다룰 수 있도록 해주는 인터페이스
- 컴퓨터의 하드웨어를 관리
- 하드웨어와 소프트웨어를 관리하는 컴퓨터 시스템의 한 부분인 실행 관리자
- 처리능력의 향상, 응답시간의 최소화, 사용 가능도의 향상, 신뢰도의 향상

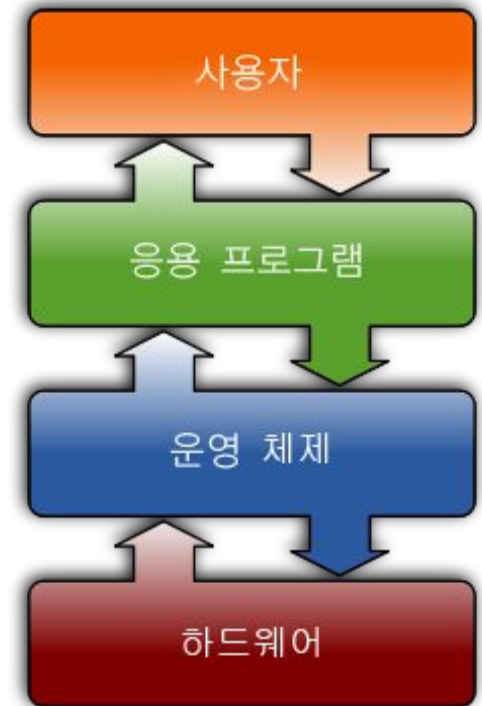


Windows 10



01 OS (운영 체제)

- OS의 기능
- 프로세서, 기억장치, 입출력장치, 파일 및 정보 등의 자원 관리
- 자원을 효율적으로 관리하기 위해 자원의 스케줄링
- 사용자와 시스템 사이 편리한 인터페이스 제공
- 하드웨어와 네트워크를 관리 및 제공
- 데이터 관리, 데이터 및 자원의 공유 기능
- 시스템의 오류를 검사 및 복구
- 자원 보호 기능
- 입/출력 보조 기능

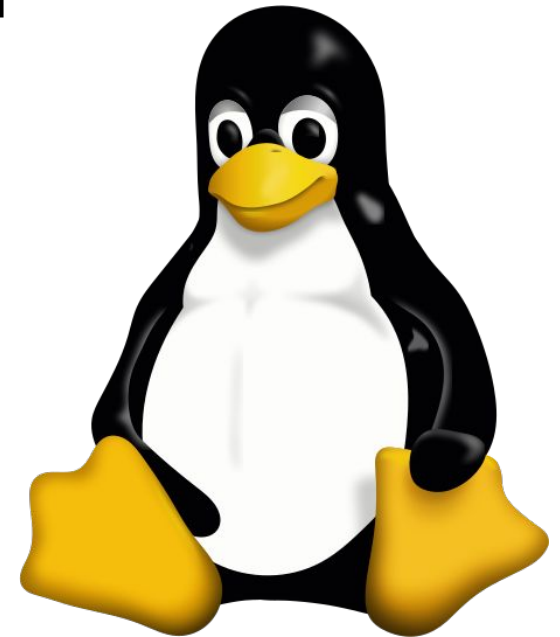


02

Linux

02 Linux

- 1991년 리누스 토르발즈가 개발한 Unix 클론
- 토르발즈가 개발한 커널에 컴파일러, 셸, 기타 응용프로그램을 조합
- Linux Is Not Unix의 준말
- Open Source 운영체제
- 현재도 활발한 개발이 진행 중
- Linux 기반의 수 많은 배포판이 존재
- 대표적으로 Ubuntu, RedHat, Suse, Debian 등



02 Linux

- Linux의 장단점

장점	단점
공개 운영체제	대부분의 언어는 영어
안정적	기술지원 부족
저사양 고효율	특정 하드웨어 지원 부족
다중 사용자 및 다중 처리 시스템	사용자의 숙련된 기술 요구
C언어로 이루어진 커널 (이식성이 좋음)	
유연성 및 확장성	

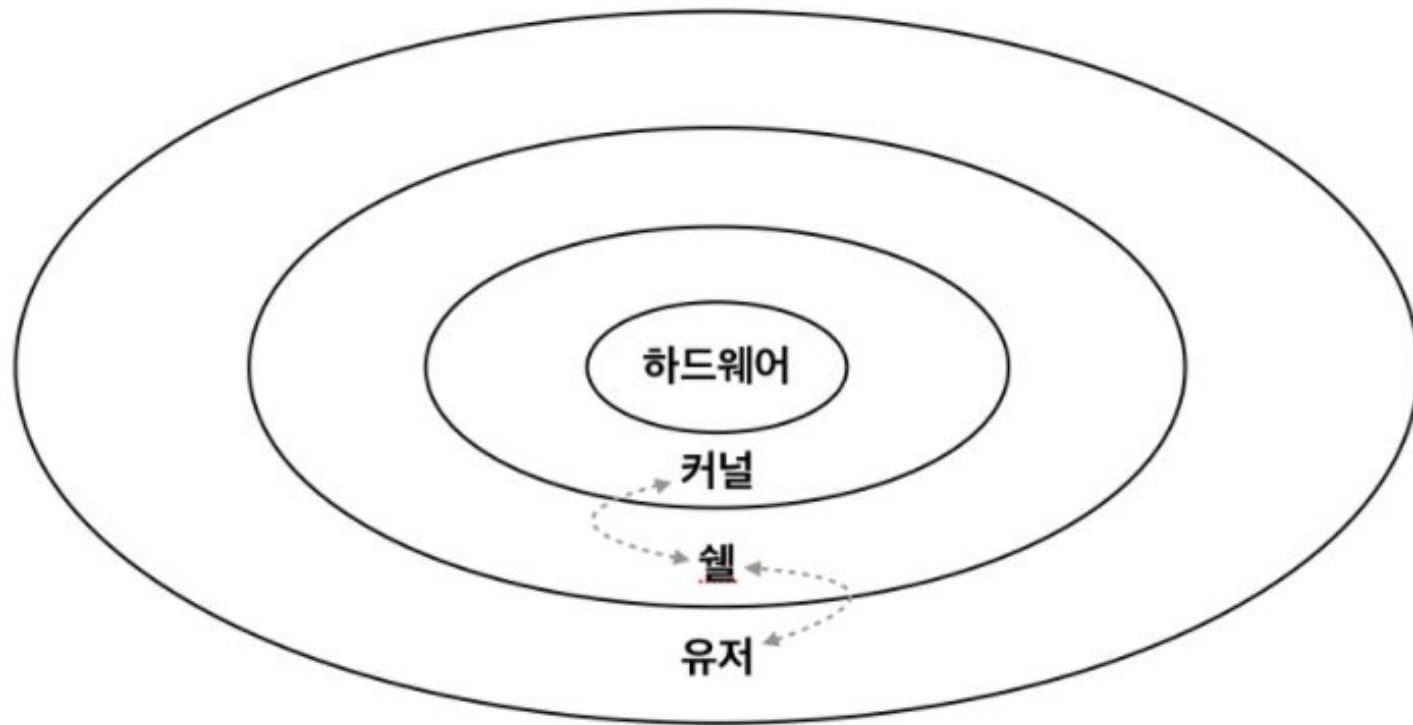
- 커널(www.kernel.org)
 - 운영체제의 핵심으로 다른 부분에 기본적인 서비스 제공
 - 커널은 부팅 될 때 메모리로 로딩됨
 - 시스템의 하드웨어, 다양한 프로세스와 메모리 등을 관리하며 시스템 제어
 - 제어하는 하드웨어 장치의 지원 여부, 정보 하드웨어 성능, 제어 코드가 존재
 - 기본 커널을 사용자가 직접 최신의 커널로 업그레이드 할 수 있음
 - 리눅스 커널은 10000줄 이상의 C언어와 1000줄 정도의 어셈블리어로 구성 → 이식성 좋음

- 셸

- 운영체제 상에서 다양한 운영체제 기능과 서비스를 구현하는 인터페이스 제공 프로그램
- 사용자가 커널, 하드웨어와 커뮤니케이션을 도움
- 명령어를 해석하여 커널로 보내주는 역할
- CLI(Command Line Interface) : 명령어 기반 인터페이스
- GUI(Graphic User Interface) : 그래픽으로 입/출력 등의 기능을 수행
- 명령어 해석기로 사용자와 OS간의 인터페이스 해석
- 키보드와 같은 단말 장치 통해 유저 입력 받아서 실행
- sh, ash, bash, csh 등 다양한 셸 존재

02 Linux

- 유저는 셸을 이용하여 명령을 전달
- 셸은 받은 명령을 커널에 전달
- 커널은 받은 명령을 통해 하드웨어를 사용



- 터미널

- 컴퓨터와 사용자간의 소통을 위한 인터페이스
- 텍스트 기반(TUI : Text-based User Interface)

단축키	설명
Ctrl + Alt + T	터미널 실행
Ctrl + Shift + C	복사하기
Ctrl + Shift + V	붙여넣기
Ctrl + Shift + T	새로운 터미널 탭 생성
Ctrl + PageUp/PageDown	이전 / 다음 터미널 탭으로 이동
Ctrl + C	현재 실행 중인 프로그램 중지
Tab	자동 완성

03

Ubuntu

03 Ubuntu

- Ubuntu 란?
- 캐노니컬과 우분투 재단에서 개발 및 배포, 유지보수를 하는 Linux 배포판
- 남아프리카 어느 부족의 언어 : 너가 있으니 나도 있다
- 자유 소프트웨어 : 자유롭게 재수정 및 재배포 가능
- Debian GNU를 기반으로 개발
- 개인 사용자와 데스크탑 환경에 최적화



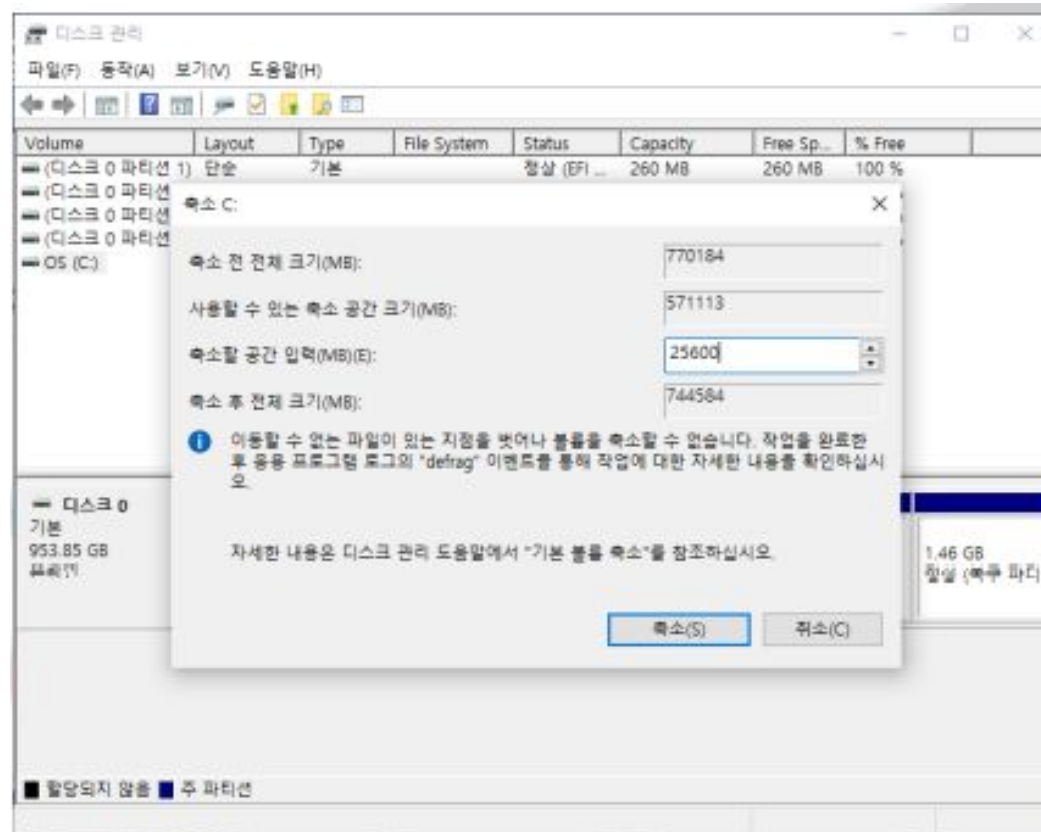
03 Ubuntu

- Ubuntu 버전
- 매년 4월 및 10월에 정규 버전 출시
- 짝수 해 4월에 LTS (Long Term Support) 버전이 출시
- 정규 버전은 6개월 마다 배포되며, 배포일 기준 9개월 간 업데이트 및 지원
- LTS 버전은 2년 마다 배포되며, 배포일 기준 최대 5년까지 업데이트 및 지원

18.04 LTS	Bionic Beaver	2018년 4월 26일	2023년 4월	4.15
18.10	Cosmic Cuttlefish	2018년 10월 18일	2019년 7월 18일	4.18
19.04	Disco Dingo	2019년 4월 18일	2020년 1월 23일	5.0
19.10	Eoan Ermine	2019년 10월 17일	2020년 7월	5.3
20.04 LTS	Focal Fossa	2020년 4월 23일	2025년 4월	5.4
21.04	Hirsute Hippo	2021년 4월 22일	2022년 1월	5.11
21.10	Impish Indri	2021년 10월 14일	2022년 7월	5.13

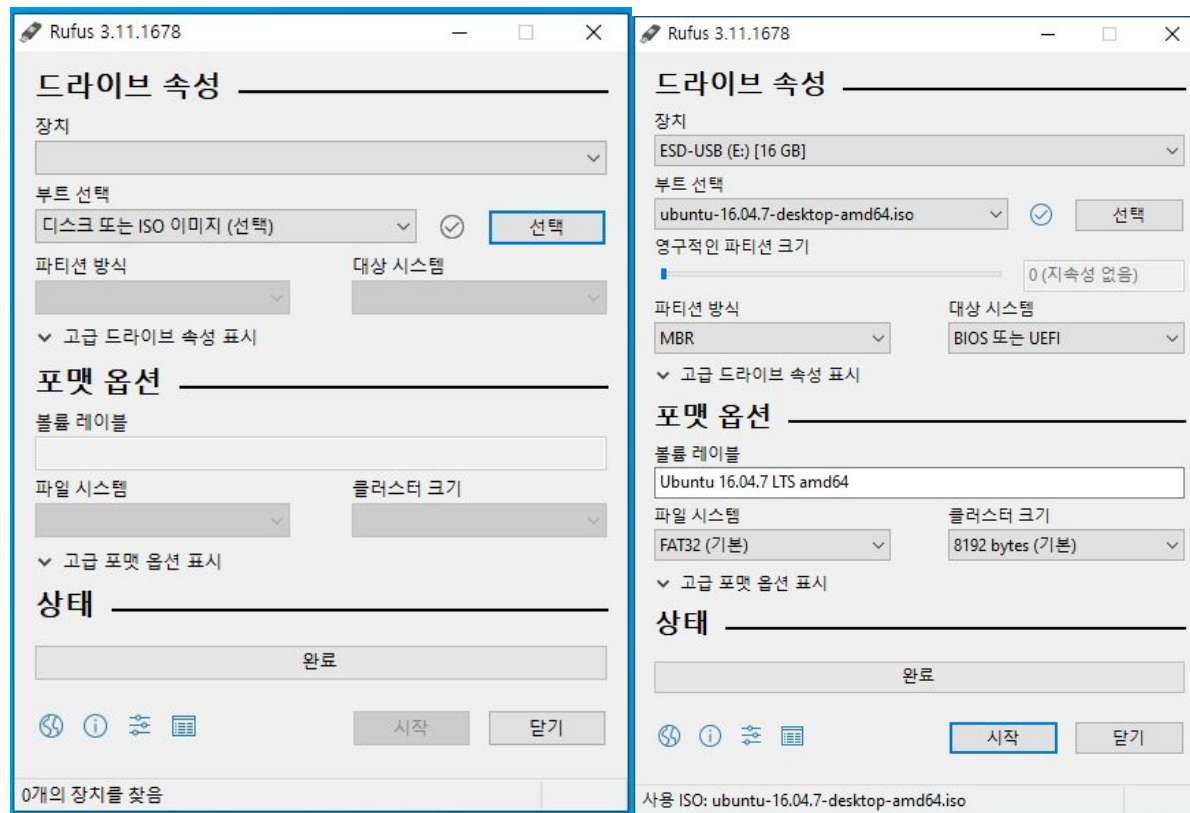
03 Ubuntu

- Ubuntu 18.04 LTS 설치
 - 파티션 할당을 먼저 진행함 > 1024 * 원하는 GB



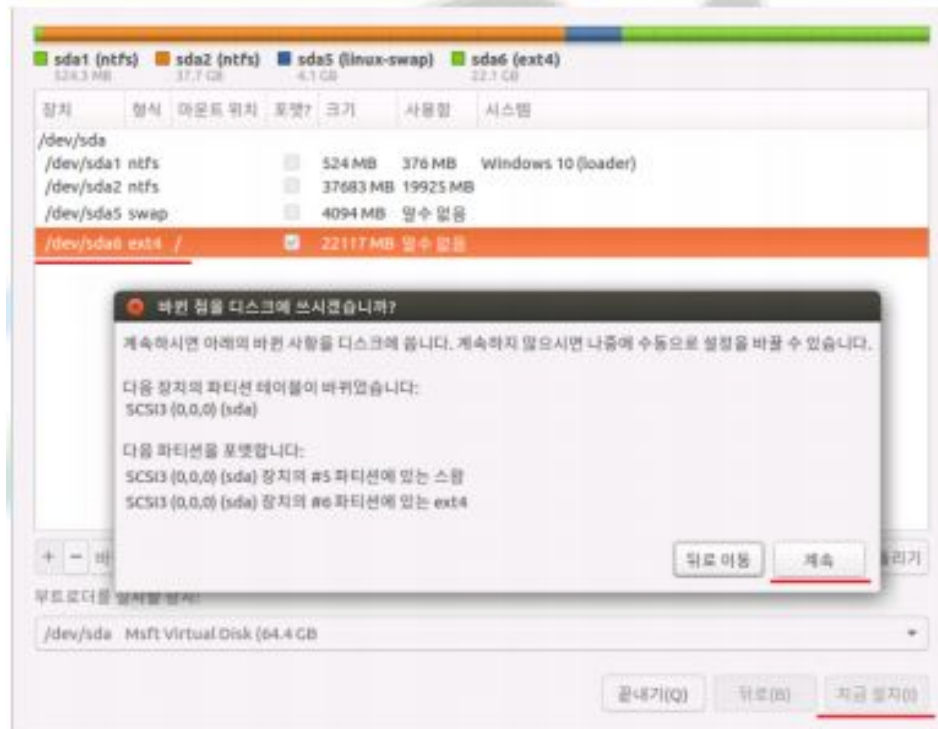
03 Ubuntu

- Ubuntu 18.04 LTS 설치
 - <https://releases.ubuntu.com/18.04.5/> : Desktop image 64bit
 - <https://rufus.ie/> : rufus 설치
 - 아래의 사진과 같이 USB에 Ubuntu를 설치



03 Ubuntu

- 재부팅을 진행하여 제조사의 로고가 나타나기 전 BIOS 창 진입
- F2 or Del 키를 누름(제조사 마다 차이가 있을 수 있음)
- Booting 순서를 USB Flash(ubuntu)를 최상위로 변경
- Ubuntu 설치 진행



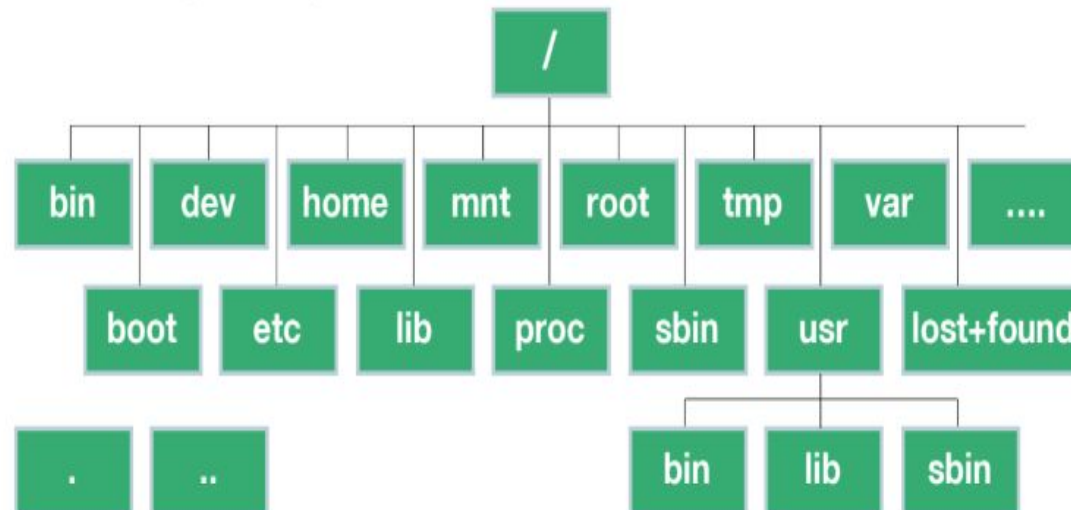
- Tip 한글 입력
 - 한글 입력 추가 / 한영키 설정
 - Setting › Region&Language › Manage installed Languages › install
 - 재부팅
 - `sudo reboot`
 - 키보드 설정
 - `Ctrl + Alt + T › ibus-setup`
 - 한글 추가
 - Input Method › Add › Korean › Hangul
 - Setting › Region&Language › Add an Input Source
 - Korean(Hangul) 추가
 - 톱니 바퀴를 눌러 설정키를 설정 할 수 있음

04

Linux Filesystem

04 Linux Filesystem

- 파일 시스템
 - 파일이나 자료를 쉽게 검색 또는 접근할 수 있도록 보관 및 조직하는 체계
 - 윈도우와 달리 최상위 폴더를 / 로 사용
 - 절대 경로 : /를 기준으로 한 전체 경로 (/home/⟨user_name⟩/catkin_ws/)
 - 상대 경로 : 현재 작업 폴더를 기준으로 한 경로 (./catkin_ws/src)
 - 자기자신을 '.', 바로 위의 디렉토리는 '..'으로 표현 (../⟨user_name⟩/catkin_ws/src)
 - ~ : Home 디렉토리를 기준 (~/ = /home/⟨user_name⟩/)



04 Linux Filesystem

- **/** : 최상위 디렉토리(모든 파일시스템의 마운트 포인터, 이름 존재)
- **/root** : 시스템 관리자의 홈 디렉토리
- **/home** : 일반 사용자들의 홈 디렉토리
- **/bin** : 기본 명령어들이 모여 있는 디렉토리
- **/boot** : 리눅스 커널이 저장되어 있는 디렉토리
- **/lib** : 자주 사용되는 라이브러리들이 저장되는 장소
- **/media** : 외부 저장매체가 마운트 되는 장소
- **/proc** : 하드웨어에 관련된 정보가 저장되는 장소
- **/etc** : 각종 환경 설정에 연관된 파일이 있는 디렉토리

04 Linux Filesystem

- `/var` : 시스템에서 사용되는 가변적인 파일들이 저장되는 장소(ex 로그)
- `/tmp` : 임시 파일들을 위한 디렉토리
- `/usr` : 시스템에 빌드하는 프로그램들의 소스나 매뉴얼 등을 포함하는 디렉토리
- `/opt` : 기본적으로 설치되는 프로그램을 제외한 소프트웨어 패키지가 존재
- `/dev` : 장치 파일, 특수 파일이 들어 있는 폴더(하드웨어를 파일로 인식)
- `/lost+found` : 파일 시스템에 문제가 발생하여 복구할 경우 문제가 되는 파일이 저장

05

Linux Command

05 Linux Command

- Bash(Bourne-Again Shell) 란?

- 현재 리눅스의 표준 셸
- Alias : 명령어 및 단축 기능
- 연산 기능
- History 기능
- 자동 이름 완성 기능
- 프롬프트 제어 기능 등이 존재
- Linux Command를 Bash에 입력하여 명령을 전달

```
wego@wego-GF63-Thin-10SCXR:~$ uname -r
5.4.0-90-generic
wego@wego-GF63-Thin-10SCXR:~$ echo Hello World
Hello World
wego@wego-GF63-Thin-10SCXR:~$ echo Hello Linux
Hello Linux
wego@wego-GF63-Thin-10SCXR:~$ █
```



BASH
THE BOURNE-AGAIN SHELL

05 Linux Command

- man (Manual)
- 특정 명령어의 사용 Manual을 표시
- `$ man <<COMMAND>>`
- 위 명령을 통해서 특정 명령어의 사용 Manual을 표시할 수 있음
- ex) `$ man ls`
`$ man chmod`
`$ man man`

05 Linux Command

- ls (list)
- 특정 위치 또는 현재 위치의 파일 목록을 조회하는 명령어
- `$ ls <<DIRECTORY>>`
- 위 명령어로 특정 위치의 파일 목록을 조회, 위치를 입력하지 않을 경우 현재 위치 조회
- 추가 Option
- `-a` : .으로 시작하는 숨겨진 파일들도 모두 표시
- `-l` : 길게 상세 정보를 표시하여 한 줄에 하나의 파일 씩 표시
- `-t` : 파일들의 생성된 최신 부터 표시
- `-F` : 유형을 나타내는 파일명을 끝에 표시(/ : 디렉토리, * : 실행파일 @ : 링크)
- ex) `$ ls -l`
`$ ls -a ~/catkin_ws`
`$ ls -lat`

- 사용자 권한
 - `$ ls -l` 명령어를 통해 리스트를 생성하면 가장 왼쪽에 다음과 같은 형태의 목록이 표시된다
 - `drwxrwxrwx` → 총 10자리로 이루어진 파일 설명
 - 가장 왼쪽은 폴더일 경우 `d`, 일반 파일을 경우 `-`로 표시
 - 오른쪽 9자리는 3자리씩 끊어서 Root, User, Others의 권한을 표시
 - Root는 최고 관리자, User : root 외의 일반 사용자, Others : 그 외의 외부인을 의미
 - 권한은 Read/Write/eXecute 의 세 가지 권한이 존재하며, 읽기, 쓰기, 실행 권한을 의미
 - 권한 변경은 파일의 소유자와 최고 관리자가 가능

05 Linux Command

- **chmod (change mode)**

- 파일의 권한을 변경하는 명령어

- `$ chmod <<options>> <<FILE_NAME>>`

- 기본 옵션으로는 `[ugoa][+-][rwx]` or `[000 ~ 777]` 을 사용하여 권한을 변경

- u 사용자 권한, g 파일을 포함하는 group 권한, o 다른 사용자의 권한, a 모두의 권한 변경을 뜻함

- +-는 권한을 주거나 뺀다는 것을 의미, rwx는 각각 읽기 쓰기 실행 권한

- 숫자 세 개로 이루어진 권한은 순서대로 user, group, others의 권한을 의미하며,

rwx가 모두 없는 것을 이진수 000, 모두 있는 것을 이진수 111로하여, 0 ~ 7까지의 숫자로 권한 변경

ex) 764 → user 7 (rwx), group 6 (rw-), others 4 (r--)

- 추가 Option

- -R : 폴더 내부의 모든 파일까지 순차적으로 권한 부여 시 사용

- ex) `$ chmod a+x <<FILE_NAME>>` `$ chmod 764 <<FILE_NAME>>` `$ chmod -R a+x <<FILE_NAME>>`

05 Linux Command

- 터미널 변수
 - 특정 Shell에서 사용하는 변수
 - Shell이 돌아가는 동안 계속 존재 및 사용 가능
 - 설정한 Shell에서만 적용됨
 - `$ echo $SHELL` : 현재 Shell 확인
 - `$ echo $HOME` : 현재 HOME폴더의 경로 확인
 - `$ export <<VARIABLE_NAME>>=<<VALUE>>` : 특정 변수를 설정하는 명령어
 - `$ unset <<VARIABLE_NAME>>` : 특정 변수를 삭제하는 명령어

05 Linux Command

- **cd (Change Directory)**
 - 특정 폴더로 이동할 때 사용하는 명령어
 - `$ cd <<DESTINATION_FOLDER>>`
 - 목표 폴더에 입력 가능한 형태는 다음과 같다
 - `/home/<<USER_NAME>>/catkin_ws/` : 절대 경로
 - `~/catkin_ws/` : 절대 경로
 - `./catkin_ws/src` : 하위 상대 경로
 - `../Download` : 상위 상대 경로

05 Linux Command

- **rm (remove)**
 - 특정 파일 또는 폴더를 삭제하는 명령어
 - `$ rm <<FILE>>`
 - 목표 폴더에 입력 가능한 형태는 `cd`와 동일함
 - 추가 가능한 옵션은 다음과 같다
 - `-r` : recursive하게 폴더 내부의 파일과 폴더까지 삭제
 - `-f` : 강제로 삭제
 - `$ rm -rf ~/catkin_ws`

05 Linux Command

- **mkdir (make directory)**
 - 특정 폴더를 추가하는 명령어
 - `$ mkdir <<FOLDER_NAME>>`
 - 목표 폴더에 입력 가능한 형태는 cd와 동일함
 - 추가 가능한 옵션은 다음과 같다
 - `-p` : 하위 폴더까지 순서대로 생성
 - `$ mkdir -p ~/catkin_ws/src/hello_world`

05 Linux Command

- pwd (print workding directory)
 - 현재 경로를 표시해주는 명령어
 - \$ pwd

05 Linux Command

- **cp (copy)**
 - 파일 또는 폴더를 복사하는 명령어
 - **\$ cp <<SOURCE_FILE>> <<DESTINATION_FILE>>**
 - SOURCE_FILE 및 DESTINATION_FILE에 입력 가능한 형태는 cd와 같음
 - 추가 가능한 옵션은 다음과 같음
 - **-r** : 파일이 아니라 폴더를 복사할 경우 내부 파일 복사 후, 폴더를 복사하는 명령어

05 Linux Command

- mv (move)
 - 파일 또는 폴더를 옮기는 명령어
 - `$ mv <<SOURCE_FILE>> <<DESTINATION_FILE>>`
 - SOURCE_FILE 및 DESTINATION_FILE에 입력 가능한 형태는 cd와 같음
 - cp와 달리 폴더도 바로 이동 가능

05 Linux Command

- 패키지 (<https://packages.ubuntu.com/>)
 - 커널 및 라이브러리 버전의 배포판 환경에 맞추어 빌드한 실행파일을 압축한 것
 - 이 실행 파일은 커널 버전 또는 설치된 OS 버전에 따라 다르게 설치 될 수 있음
 - 특정 패키지의 설치를 위해서 미리 설치되어야하는 의존성 패키지가 존재
 - \$ sudo apt install <<PACKAGE_NAME>> : 특정 패키지를 설치하는 명령어
 - \$ sudo apt update : 저장소를 다시 불러와서 업그레이드 가능한 패키지를 확인
 - \$ sudo apt upgrade : 업그레이드 가능한 패키지를 모두 업그레이드
 - \$ apt list --upgradable : 업그레이드 가능한 패키지 리스트 확인
 - \$ sudo apt install --only-upgrade <<PACKAGE_NAME>> : 특정 패키지만 업그레이드

06

Etc

- Nano 편집기
- 터미널에서 사용 가능한 편집기 중 사용 방법이 쉬운 편에 속하는 편집기
- \$ nano <<FILE_NAME>> 으로 편집을 시작할 수 있음
- 실행 중 아래에 사용 방법이 표시되며, 저장을 위해서는 Ctrl + X 입력하여 저장할 파일의 이름을 정하면 저장 가능



The screenshot shows the GNU nano 2.9.3 text editor interface. The title bar at the top indicates the editor version and the file name 'hello_world.c'. The main editing area contains the following C code:

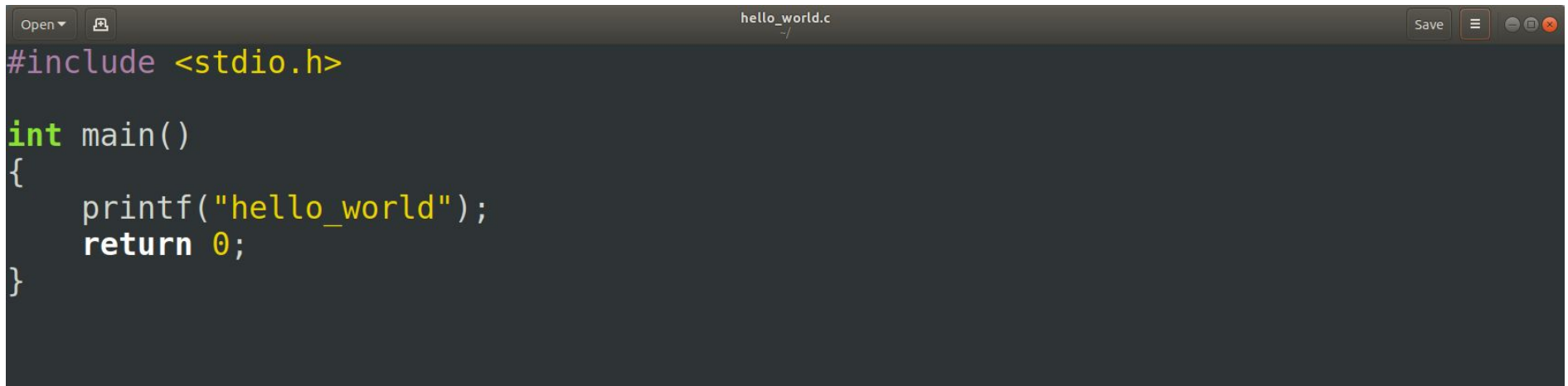
```
#include <stdio.h>

int main()
{
    printf("hello_world");
    return 0;
}
```

At the bottom of the editor, a status bar displays various keyboard shortcuts for navigation and editing. A small box above the status bar indicates 'Read 7 lines'.

Shortcut	Action
^G	Get Help
^O	Write Out
^W	Where Is
^K	Cut Text
^J	Justify
^C	Cur Pos
^X	Exit
^R	Read File
^\	Replace
^U	Uncut Text
^T	To Spell
^_	Go To Line

- Gedit 편집기
 - GUI환경에서만 사용 가능
 - 일반적으로 사용 가능한 단축키를 거의 다 사용 가능
 - Python 코드 작성 시. Tab으로 공백 생성과 Space로 공백 생성이 다른 문제가 있음

A screenshot of the Gedit text editor window. The title bar shows 'hello_world.c'. The menu bar includes 'Open', a file icon, and 'Save'. The editor area contains the following C code:

```
#include <stdio.h>

int main()
{
    printf("hello_world");
    return 0;
}
```


- Visual Studio Code
 - GUI환경에서만 사용 가능
 - 일반적으로 사용 가능한 단축키를 거의 다 사용 가능
 - 일반적으로 사용하기에 좋은 Editor
- 설치 방법
 - `$ sudo apt-get install curl`
 - `$ sudo sh -c 'curl https://packages.microsoft.com/keys/microsoft.asc | gpg --dearmor > /etc/apt/trusted.gpg.d/microsoft.gpg'`
 - `$ sudo sh -c 'echo "deb [arch=amd64] https://packages.microsoft.com/repos/vscode stable main" > /etc/apt/sources.list.d/vscode.list'`
 - `$ sudo apt update`
 - `$ sudo apt install code`

- **SSH**
 - Secure Shell Protocol : 서로 통신 할 때 보안적으로 안전하게 통신을 위한 프로토콜
 - 데이터 전송, 원격 제어가 대표적인 사용 예
- **설치 방법**
 - `$ sudo apt install ssh`
- **접속 방법**
 - `$ ssh <<USER_NAME>>>@<<IP_ADDRESS>>`
 - USER_NAME과 IP_ADDRESS를 설정하여 접속하여 사용 가능 (UI는 없음)



WeGo Robotics

Tel. 031 – 229 – 3553

Fax. 031 – 229 – 3554



제품 문의: go.sales@wego-robotics.com

기술 문의: go.support@wego-robotics.com