

# How to use Shell?

**A Basic CS Skill, ABC Winter School**

**박원**

# Objective

- **BASH** 에 대해 알 수 있다.
- 주로 사용하는 **shell 명령어**를 이해하고, 사용할 수 있다.

# About OS

- 컴퓨터에는...
  - cpu, gpu, ram 등의 구성요소
- 어떻게 하면 이 장치들을 잘 사용할 수 있을까요?
- 자원을 효과적으로 사용할 수 있도록 관리해야한다!
  - 메모리 영역 할당, cpu 연산 스케줄링 등
- 이걸 해주는게 바로 OS

# What is the shell?

- 요즘의 컴퓨터는 다양한 interface를 통해 명령을 할 수 있게 되어있어요.
  - 이걸 보여주고, 받아들이는 것도 OS의 역할!
- 우리는 보통 GUI(Graphic User Interface)를 통해 컴퓨터와 소통을 해요
  - 특이하게는 Voice Interface(Siri, 빅스비 등등)를 사용하기도 하고
  - AR/VR도 있어요!

# What is the shell?

- 다양한 인터페이스 중에서도 가장 먼저 사용 된 것은
  - CLI(Command-Line Interface) 입니다.
- 명령어를 모두 알고 있다는 가정하에, 가장 빠르게 컴퓨터에게 명령을 내릴 수 있어요.
- 하지만 직관력이 부족하고, 진입장벽이 높아요.

# Component of Shell

- 셸 화면이 어떻게 나타나는지를 먼저 확인할게요
  - 우리는 oh-my-zsh와 powerlevel10k 테마를 사용하고 있기 때문에 조금 달라요!

```
[abc@unist ~]$ command arg1 arg2
```

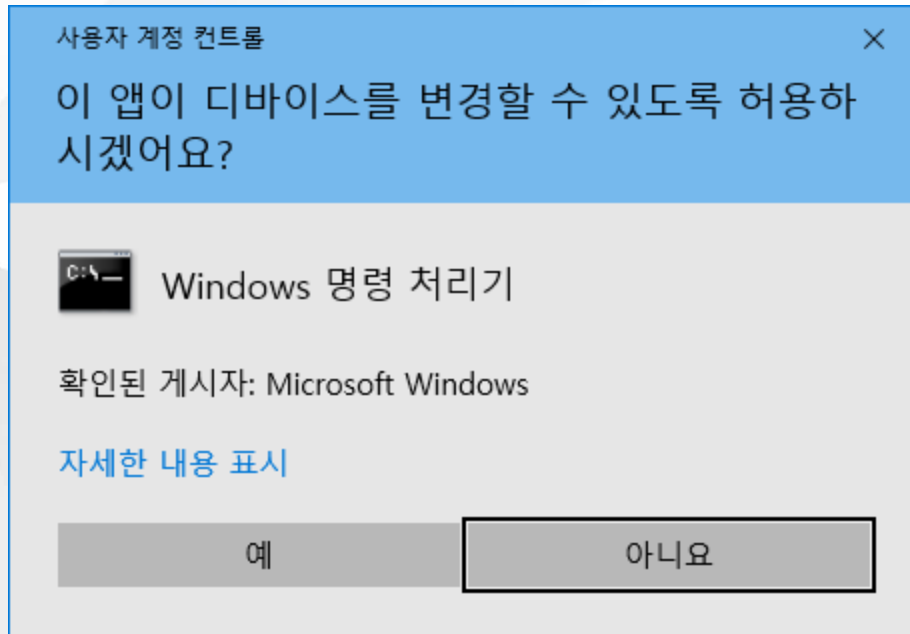
- abc - 지금 세션 혹은 터미널에 로그인한 계정 명
- unist - 서버 또는 하드웨어 장치의 이름
- ~ - 현재 위치한 디렉토리의 경로

# Component of Shell

```
[abc@unist ~]$ command arg1 arg2
```

- \$ - 현재 권한이 일반 사용자 권한임을 의미해요
  - # 이면 sudo 권한임을 의미해요 자세한건 뒤에서 설명할 예정!
- command - 명령어 입니다. 사실 각 명령어는 모두 하나의 프로그램으로써 동작해요!
- arg - 명령을 수행할 때 들어가는 argument들이에요!
- Shell은 이 명령어를 알아서 파싱해서 실행시켜줘요!

# Sudo?



- 익숙한 화면이죠?



# Sudo?

- 아무런 경고 없이 실행가능한 프로그램이 있고, 그렇지 않은 프로그램이 있어요.
- 후자의 경우 시스템을 변경하거나, 혹은 중요한 영향을 미치는 경우에 그래요.
- 이런 경우, 우리는 시스템에서 관리자 권한을 가진 계정의 허락을 받아야합니다.
- 이 관리자 권한을 sudo 권한이라고 하고, cli에서는 `sudo command` 이런 식으로 명령을 입력해요!

# Directory

- 리눅스도 macOS도 UNIX 파일시스템을 따르고 있어요
- 윈도우는 익숙할테고, UNIX(unix-like)를 사용하기로 했으니, 이것만 살펴볼게요
- UNIX의 최상위 폴더는 **root** 폴더예요.
- 이 **root** 아래에 모든 파일이 존재합니다.

# Directory

- `/dev` - 하드웨어가 접근하는 폴더입니다. 보통은 들어갈일이 없어요
- `/mnt` - 마운트한 드라이브가 여기에 있어요. C, D 드라이브 같은거에요.
- `/lib` - 라이브러리 등이 담겨있습니다.
- `/usr` - 유저가 설치한 프로그램이 들어가요.
- `/bin` - 시스템 프로그램이 들어가요. ex: `/bin/echo`

# Home Directory?

- 각각의 유저는 고유한 폴더를 가지고 있어요.
- 이를 home directory라고 부릅니다.
  - `~` 으로 표현되어요!
- 경로는 `/home/<username>` 이에요.

# Directory

- 절대경로 vs 상대경로
- 절대경로는 `root directory` 부터의 경로입니다.
- 상대경로는 root가 아닌 어떤 특정 directory부터의 경로입니다.
  - `~` 는 아까 설명했듯 home입니다.
  - `.` 은 현재 위치를 가리킵니다.
  - `..` 은 상위 디렉토리를 가리킵니다.

# Command - pwd

- Print Working Directory
- 지금 위치하고 있는 곳의 절대경로를 알고 싶을때 사용해요

```
[abc@unist ~]$ pwd  
/home/abc
```

# Command - echo

```
echo [arg]
```

- echo는 `arg`에 들어간 내용을 그대로 출력해요!

```
[abc@unist ~]$ echo hello!  
hello!
```

# Command - ls

```
ls [options]
```

- ls는 현재 directory에 있는 파일 및 directory를 출력해요.

```
[abc@unist ~]$ ls  
abc unist hello.txt
```



# Command - cd

```
cd [directory name]
```

- cd는 다른 위치로 이동해요.

```
[abc@unist ~]$ cd unist  
[abc@unist ~/unist]$
```

# Command - cp

```
cp <target> <dest>
```

- 파일을 복사해요.
- 폴더의 경우에는 `cp -r <> <>` 으로 사용하면 됩니다

```
[abc@unist ~/unist]$ ls
hello.txt
[abc@unist ~]$ cd hello.txt hello2.txt
[abc@unist ~/unist]$ ls
hello.txt hello2.txt
```

# Command - mv

```
cp <target> <dest>
```

- 파일을 옮겨요.
- 폴더의 경우에는 `mv -r <> <>` 으로 사용하면 됩니다

```
[abc@unist ~/unist]$ ls
hello.txt
[abc@unist ~]$ mv hello.txt hello2.txt
[abc@unist ~/unist]$ ls
hello2.txt
```

# Command - rm

```
rm <target>
```

- 파일을 지워요.
- 폴더의 경우에는 `rm -r <target>` 으로 사용하면 됩니다.

```
[abc@unist ~/unist]$ ls  
hello.txt  
[abc@unist ~]$ rm hello.txt  
[abc@unist ~/unist]$ ls
```

# Command - mkdir

```
mkdir <target>
```

- 폴더를 생성해요

```
[abc@unist ~/unist]$ ls
```

```
[abc@unist ~]$ mkdir abc
```

```
[abc@unist ~/unist]$ ls
```

```
abc
```

# Command - ssh

```
ssh <username>@<server> -p <port>
```

- ssh는 원격 컴퓨터와의 통신을 위한 보안 프로토콜이에요. (Secure SHell)
- <username>@<server address> 에 접속을 하고, <port>로 접속하라는 의미예요.
- 일반적으로는 **-p** 옵션을 굳이 달지 않아도 괜찮아요.
- 명령을 하게되면 암호를 입력하게 되는데, 암호는 보이지 않습니다.
- 서버가 생기면 한번 해보세요...!

# Command - scp

```
scp <filename> <username>@<server>:<target>
```

- <username>@<server> 의 <target> 위치에 <filename>을 복사해서 전송합니다.
- 폴더의 경우에는 **-r** 옵션이 필요해요!
- cp와 ssh를 합친거라고 생각하면 편해요.
- 서버가 생기면 한번 해보세요...!

# Command - clear

- 화면을 깨끗하게 지워줘요!



# Command - exit

- shell을 종료해요

# apt

- Only for Ubuntu!!
- 우분투 패키지 관리 툴이에요  
`apt install <package_name>`
- 위 명령어로 설치할 수 있어요.

```
apt update  
apt install python3
```

# Questions?

# Next

## Shell Tools and Scripting