

Basic CLI  
Commands

# Basic Linux 명령어



# Linux CLI

- CLI : Command Line Interface
  - 명령어를 이용하여 컴퓨터를 제어하는 방식
  - GUI에 비해 메모리 등을 덜 차지합니다.
  - 무거운 (복잡한) 컴퓨터 작업을 하기에 용이합니다.

```
> neofetch
neofetch: command not installed, but was located via Nix.
Would you like to run neofetch from Nix? [Yn]: Y
Add 'neofetch' to repl.it.nix if you want to install 'neofetch' in this repl.
/nix/store/smbnd3h1r86f7yfx8kh5hpr0lspfdssz-neofetch-unstable-2020-11-26
.-/+oossssoo+/-.
`:+ssssssssssssssss+:`
-+ssssssssssssssssyyss+-
.oSSsssssssssssssdMMMNysssO.
/SSssssssshdmmNNmyNMMMMhsssss/
+ssssssshmydMMMMMMNdddyssssss+
/SSssssshNMMMyhhyyyhmNMMNhsssss/
.sSSssssdMMMNhssssssshNMMMdssssss.
+ssshhhyNMMNysssssssssyNMMMyssssss+
ossyNMMMNyMMhssssssssshmmhssssssso
ossyNMMMNyMMhssssssssshmmhssssssso
+ssshhhyNMMNysssssssssyNMMMyssssss+
.sSSssssdMMMNhssssssshNMMMdssssss.
/SSssssshNMMMyhhyyyhdNMMNhsssss/
+sssssssdmydMMMMMMNdddyssssss+
/SSssssssshdmmNNmyNMMMMhsssss/
.oSSsssssssssssssdMMMNysssO.
-+ssssssssssssssyyss+-
`:+ssssssssssssss+:`
.-/+oossssoo+/-.

runner@aac6eeb83806
-----
OS: Ubuntu 20.04.2 LTS x86_64
Host: Google Compute Engine
Kernel: 5.13.0-1019-gcp
Uptime: 10 hours, 39 mins
Packages: 141 (apt), 42 (nix-user)
Shell: bash 4.4.23
Resolution: 801x622
Terminal: pid1
CPU: AMD EPYC 7B12 (8) @ 2.249GHz
Memory: 19402MiB / 64311MiB
```

# 메뉴얼

- `man [command]`
  - Manual
  - `h` : 사용법 확인
  - `q` : 나가기
  - `<↑>` / `<↓>` : 한 줄 씩 이동
  - `<Space Bar>` : 한 페이지 이동
  - `/[keyword]` : 검색
- `[command] --help`
  - 간단한 사용법을 알 수 있다.
  - 자세한 사용법은 `man`을 사용

```
$ man man↵
man(1)  General Commands Manual  man(1)

NAME

    man - format and display the on-line
manual pages

SYNOPSIS

    man [-acdfFhkKtwW] [--path] [-m system]
[-p string] [-C config_file] [-M pathlist]
[-P pager] [-B browser] [-H htmlpager] [-S
section_list] [section] name ...
```

# File System

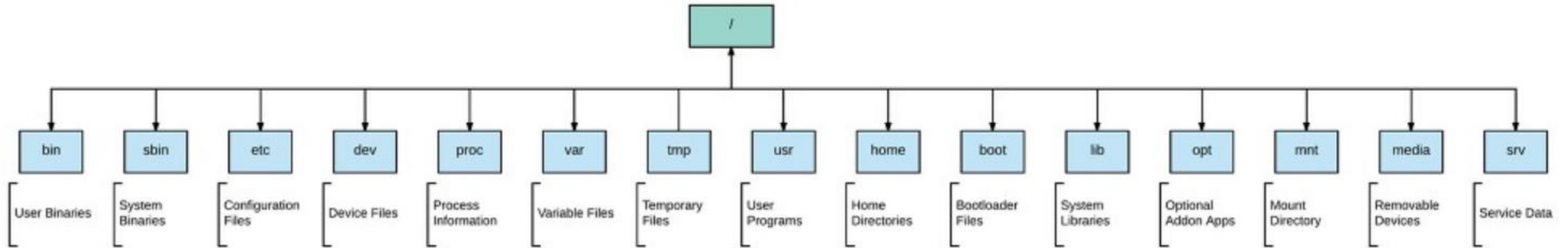


# File System

- File Tree

- / 리눅스 파일 시스템의 최상위
- /bin Binarise, 부팅 실행파일 & 다양한 명령어
- /tmp Temporary, 임시파일 저장소
- /usr User, 각종 응용 프로그램의 설치 장소
- /home Home, 계정 사용자들의 홈 디렉토리, 서비스 디렉토리
- /boot Boot, 부팅에 꼭 필요한 지원 파일들
- /lib Library, 각종 프로그램의 라이브러리들

# File System



- 처음 터미널을 통해 로그인 시,  
"/home/[username] (= ~)"에서 시작
- 위와 같은 트리구조를 옮기며 작업

현재 Directory:	.
상위 Directory:	..
루트 Directory:	/
홈 Directory:	~
이전 Directory:	-
자동 완성:	<Tab>

# 탐색하기

- pwd
  - Print Working Directory
- ls (option) (path)
  - List Segments
  - -a : all 디렉토리 안에 있는 모든 파일과 디렉토리를 보여준다.
  - -l : list 파일에 대한 정보를 자세하게 보여준다.
- cd [path]
  - Change Directory

```
$ pwd
/home/jtjun
$ ls
main main.c my_dir
$ ls -a my_dir
.  ..  my_file
$ cd my_dir
$ ls -l
-rw-r--r- 1 root root 0 Mar 21 my_file
$ cd ..
$ ls
main main.c my_dir
```

# 생성 & 삭제

- `mkdir [targets]...`
  - Make Directory
- `touch [targets]...`
  - Touch
  - Time Stamp 업데이트 or 새 파일 생성
- `rm (opt) [targets]...`
  - Remove
  - `-f` : force 강제로 삭제
  - `-r` : recursive 디렉토리 삭제
  - `-d` : 비어 있는 디렉토리만
  - `-v` : 삭제되는 대상 정보 출력

```
$ mkdir new_dir↵
$ touch new_file↵
$ ls↵
main main.c my_dir new_dir new_file
$ rm new_file↵
$ rm -r new_dir↵
$ ls↵
main main.c my_dir
```



# 옮기기

- mv [source] [destination]
  - Move
  - Directory 또는 File 을 이동 (rename)
- cp (opt) [src] [dst]
  - Copy
  - -f : 이미 존재하는 경우, 강제 덮어쓰기
  - -i : 이미 존재하는 경우, 덮어쓸지 묻기
  - -b : 이미 존재하는 경우, 백업
  - -r : 하위까지 모두 복사
  - -v : 복사 진행상태 출력

```
$ mv main exe↵
$ ls↵
exe main.c my_dir
$ cp exe exe2↵
$ cp -r my_dir new_dir↵
$ ls↵
exe exe2 main.c my_dir new_dir
$ ls my_dir↵
my_file
$ ls new_dir↵
my_file
```

# Printing



# 출력하기

- `cat (opt) [targets]...`
  - Catenate
  - `-n` : 번호 출력
- `echo "[msg]" (> [target])`
  - Msg 를 콘솔에 출력
  - `> [target]` 을 입력한 경우, 해당 타겟에 msg를 덮어 씌움
  - `>> [target]` 을 입력한 경우, 해당 타겟 내용에 msg를 뒤에 추가함

```
$ cat main.c↵
# include <stdio.h>

int main() {
    printf("Hello World\n");
    return 0;
}
$ echo "Hello World"↵
Hello World
$ echo "Hi" > hello.txt↵
$ cat hello.txt↵
Hi
```

# 출력하기

- `more [target]`
  - 한 페이지만큼만 출력,  
더 불러오며 내용을 확인
  - 지나간 내용을 볼 수 없음
  - <Enter> : 아래로 1행 이동
  - <Space Bar> : 아래로 1페이지 이동
- `less [target]`
  - 한 페이지만큼만 출력,  
more과 동일하게 불러오며 확인
  - 지나간 내용을 볼 수 있음
  - <↑> : 위로 1행 이동
  - <↓> : 아래로 1행 이동

```
$ more my_dir/my_file↓
Once a farmer had a dog,
And Bingo was his name oh!
B-I-N-G-O
B-I-N-G-O
B-I-N-G-O
and Bingo was his name oh.
Once a farmer had a dog,
And Bingo was his name oh!
B-I-N-G-O
B-I-N-G-O
--More-- (32%)
```

# 출력하기

- [command] #
  - 주석 달기
- clear
  - 콘솔 화면 비우기

```
$ clear # clear the screen.↵
```

# Process



# Process

- 프로그램이란?
  - 어떤 작업을 위해 실행이 가능한 '정적 상태'의 파일
  - 명령어와 데이터들의 집합
- 프로세스란?
  - 실행 중인 '동적 상태'의 프로그램
  - PID : 각 프로세스를 식별할 수 있는 번호
  - State : 실행, 준비, 블록 (상태 전이)
  - Context : 실행상태의 문맥, 레지스터 내용
  - Memory : 각각 독립된 메모리 영역

# Process 관리하기

- ps (opt)
  - Process
  - -p [pid] : 특정 프로세스
  - -r : 현재 실행 중인 프로세스
- kill (opt) [pid]
  - -s [signal] : 시그널 보내기
  - -l : 가능한 시그널 목록 확인
- <Ctrl> + Z
  - 현재 프로세스 '일시정지'
- <Ctrl> + C
  - 현재 프로세스 '강제종료'

```
$ ps
  PID TTY          TIME CMD
   19 pts/0        00:00:00 bash
 2788 pts/0        00:00:00 exe
 2846 pts/0        00:00:00 ps
$ kill -s KILL 2788
$ ps
  PID TTY          TIME CMD
   19 pts/0        00:00:00 bash
 2878 pts/0        00:00:00 ps
[1]+  Killed                  ./exe
```



# Background 실행하기

- [command] &
  - Command를 백그라운드에서 실행
- jobs
  - 백그라운드에서 실행 중인 작업 확인
- fg %[job\_id]
  - Background -> Foreground
- bg %[job\_id]
  - Foreground -> Background
- nohub [command] &
  - 터미널이 닫혀도 계속 실행

```
$ ./exe &
[1] 48
$ jobs
[1]+  Running                  ./exe &
$ fg %1
./exe
^Z
[1]+  Stopped                  ./exe
$ bg %1
[1]+ ./exe &
$ jobs
[1]+  Running                  ./exe &
```

Others



# 권한

- `sudo [command]`
  - Substitute (super) User DO
  - 다른 사용자 권한으로 실행 (관리자 권한으로 실행)
  - `chmod` 권한을 무시할 수는 없음
- `chmod [mode] [target]`
  - Change Mode
  - 소유자 u, 그룹 g, 그 외 o, 모든 a
  - 추가 +, 제거 -, 지정 =
  - 읽기 r, 쓰기 w, 실행 x
- `chown [user:group] [target]`
  - Change Owner

```
$ chmod a-x main↵
$ ./main↵
bash: ./main: Permission denied
$ sudo ./main↵
[sudo] password for jtjun:
sudo ./main: command not found
$ chmod a+x main↵
$ ./main↵
Hello World!
$ chown root:root main↵
$ ls -l main↵
-rwxr-xr-x 1 root root 54136 Mar 21 main
```

# 원격 접속

- ssh [dst]
  - Secure Shell
  - [dst] := [id]@[ip]
  - 네트워크를 통해 다른 Linux 서버로 접속하는 암호화 지원 프로토콜
  - exit : 접속 해제
- scp (opt) [src] [dst]
  - Secure Copy
  - [dst] := [id]@[ip]:[path]
  - -r : 디렉토리 내 모든 대상 복사
  - -p : 권한 속성 유지 복사
  - -P [port] : 포트 지정 복사

```
$ ssh jtjun@ssdlab.snu.ac.kr↵
jtjun@ssdlab.snu.ac.kr's password:
Welcome to Ubuntu 16.04.6 LTS

 * Documentation:      https://help.ubuntu.com/
 * Management:        https://landscape.canonical.com/
 * Support:            https://ubuntu.com/support

Last login: Mon Mar 21 00:00:00 2022
jtjun@1a033ae57c7:~#
```

# 통용 옵션

- `[command] > [target]`
  - Command 의 실행으로 콘솔에 출력되는 내용을 target에 저장
- `[command] < [target]`
  - Command 의 실행으로 콘솔에서 입력할 내용을 target의 내용으로 입력
- `[command0] | [command1] | ...`
  - Pipeline
  - 여러 명령어를 순차적으로 실행

```
$ ./main↵
World
Hello World!
$ ./main > output.txt↵
Linux
$ cat output.txt↵
Hello Linux!
$ cat input.txt↵
CLI
$ ./main < input.txt↵
Hello CLI!
```

감사합니다.

