

Linux 入門

- [\[Linux #1\] Basic Command](#)
- [\[Linux #2\] ファイルの操作](#)
- [\[Linux #3\] cmd でのファイルの操作](#)
- [\[Linux #4\] vi nano でのファイルの操作](#)

[Linux #1] Basic Command

키보드: `Tab` 키

키보드: `Shift + INS` 키 복사

1. ls (List)

ls : 현재 디렉토리 내의 파일과 디렉토리 목록을 표시합니다.

ls -l : 파일과 디렉토리에 대한 자세한 정보를 표시합니다.

ls -a : 숨겨진 파일과 디렉토리를 포함하여 모든 항목을 표시합니다.

image.png

2. cd (Change Directory)

cd : 현재 디렉토리를 변경합니다.

cd Documents : 'Documents' 디렉토리로 이동합니다.

cd .. : 상위 디렉토리로 이동합니다. (루트, /)

3. pwd (Print Working Directory)

pwd : 현재 작업 중인 디렉토리의 절대 경로를 출력합니다.

image.png

4. mkdir (Make Directory)

mkdir 创建目录(文件夹)

mkdir new_folder : 创建名为 'new_folder' 的目录

mkdir -p folder1/folder2 : 'folder1' 和 'folder2' 创建. -p 创建父目录, 如果不存在则创建.

5. rmdir (Remove Directory)

rmdir 删除目录

rmdir old_folder : 'old_folder' 删除

rmdir 只能删除空目录. 如果目录不是空的, 则无法删除.

删除非空目录使用 rm -r 命令.

6. rm (Remove)

rm 删除文件

rm file.txt : 'file.txt' 删除

rm -r folder : 'folder' 删除及其子目录

rm 只能删除文件, 不能删除目录. 删除目录使用 rm -r.

7. touch

touch 创建空文件, 或更新文件(如果文件存在)的修改时间.

touch new_file.txt : 'new_file.txt' 创建空文件. 如果文件存在, 则更新其修改时间.

touch 只能创建空文件或更新现有文件的修改时间, 不能创建目录.

8. cp (Copy)

cp 是 Linux 中用于复制文件的命令。

cp source.txt destination.txt : 'source.txt' 复制到 'destination.txt'。

cp -r source_dir destination_dir : 'source_dir' 递归复制到 'destination_dir'。

9. mv (Move)

mv 是 Linux 中用于移动或重命名文件的命令。

mv old_name.txt new_name.txt : 'old_name.txt' 重命名为 'new_name.txt'。

mv file.txt /path/to/directory/ : 'file.txt' 移动到指定目录。

mv 还可以用于移动目录。例如，将目录 'old_dir' 移动到 'new_dir'。

mv 命令在移动文件或目录时，会保留文件的权限和属性。

10. cat (Concatenate)

cat 是 Linux 中用于连接文件的命令。

cat file.txt : 显示 'file.txt' 的内容。

cat file1.txt file2.txt > combined.txt : 'file1.txt' 和 'file2.txt' 的内容合并到 'combined.txt'。

11. chmod (Change Mode)

chmod 是 Linux 中用于更改文件权限的命令。

`chmod 755 file.sh` : 'file.sh' 실행 권한 부여, 읽기, 쓰기 권한 부여, 실행 권한 부여.

`chmod u+x file.sh` : 'file.sh' 사용자 권한 부여.

이 명령어는 파일의 권한을 변경하는 데 사용됩니다.

12. grep (Global Regular Expression Print)

grep은 텍스트 파일에서 패턴을 검색하고 일치하는 줄을 출력하는 도구입니다.

`grep "text" file.txt` : 'file.txt'에서 "text" 패턴을 검색하고 일치하는 줄을 출력.

`grep -r "text" .` : 현재 디렉토리에서 "text" 패턴을 검색하고 일치하는 줄을 출력.

grep은 정규 표현식을 사용하여 패턴을 검색할 수 있으며, 여러 파일을 검색할 수도 있습니다.

13. echo

echo는 텍스트 줄을 출력하는 명령어입니다. 기본 사용법은 다음과 같습니다.

`echo "Hello World"` : "Hello World"를 출력.

`echo $HOME` : 'HOME' 환경 변수의 값을 출력.

`echo "some text" > file.txt` : "Some text"를 'file.txt'에 저장.

echo는 여러 줄을 출력할 수 있으며, 백슬래시를 사용하여 줄을 연결할 수 있습니다.

14. man (Manual)

man은 시스템의 다양한 명령어와 파일에 대한 설명을 제공하는 도구입니다.

`man ls` : 'ls' 명령어의 사용법을 출력.

15. sudo (SuperUser DO)

root (superuser) can run any command. root can do anything, root can delete files, root can change permissions, root can change the system.

```
sudo apt-get update : □□ □□□ □□□ □.
```

16. find

```
find . -name "file.txt" : ☐ ☐ ☐ ☐ ☐ ☐ 'file.txt' ☐ ☐ ☐.
```

```
find / -type d -name "config" :  0 000000 'config' 0 00 0000 00.
```

17. cp (copy)

- cp source_file destination: .

18. cat (Concatenate)

cat , .

```
cat file.txt: file.txt
```

cat file1.txt file2.txt:

19. cat (Concatenate)

cat _____, _____.

```
cat file.txt: file.txt [ ] [ ] [ ].
```

cat file1.txt file2.txt: 一行一行地显示文件内容。

20. vi(Visual Editor) 编辑器

vi 是 Linux 系统中常用的文本编辑器。它可以在文本文件中对文本进行编辑。启动 vi 编辑器的命令如下：

vi file.txt: 打开文件 file.txt 进行编辑。

i 进入插入模式

Esc 退出插入模式

:w 保存文件

:q 退出 vi

:wq 保存并退出

:q! 强制退出

[Linux #2] 環境構築

環境構築, 1. 環境構築, Docker compose 環境構築?

```
sudo apt update
sudo apt install docker.io
sudo apt install docker-compose
```

環境構築 1. 環境構築

```
dainwiki@DESKTOP-4OPVAQ5:~$ docker-compose up
ERROR: yaml.scanner.ScannerError: mapping values are not allowed here
in "./docker-compose.yml", line 6, column 31
```

ERROR: yaml.scanner.ScannerError: mapping values are not allowed here in "./docker-compose.yml", line 6, column 31

yaml 6 31 行目のエラー

```
vi docker-compose.yml
i
:6
```

image.png

6 31 行目のエラー ()

yaml Validator 環境構築 Valid yaml 環境構築

環境構築

1. .env 環境構築

`.env` `DB_HOST=localhost` `DB_NAME=bookstack` `DB_USER=bookstack` `DB_PASS=bookstack` `DB_PORT=5432`

```
vi .env
```

`DB_HOST=localhost` `DB_NAME=bookstack`

2. `bookstack` `bookstack`

`bookstack` `bookstack`

```
dainwiki@DESKTOP-4OPVAQ5:~$ docker --version
Docker version 24.0.7, build 24.0.7-0ubuntu2~22.04.1
```

`bookstack` `bookstack`

```
dainwiki@DESKTOP-4OPVAQ5:~$ sudo chmod -R 755 /home/dainwiki/bookstack_data
dainwiki@DESKTOP-4OPVAQ5:~$ sudo chmod -R 755 /home/dainwiki/bookstack_db
```

3. `bookstack` `bookstack`

```
sudo apt install net-tools
netstat -ano
```

[Linux #3] cmd



이제 IP를 지정해서 서버를 실행해 보겠습니다.

이제 서버를 실행하고, 클라이언트가 접속하면 어떻게 되는지 확인해 보겠습니다.

cmd에서 netstat -an

```
netstat -an | find "8080"
```

```
netstat -an | find "80"
```

image.png

이제 서버가 LISTENING 상태에 있는지 확인해 보겠습니다.

TCP	0.0.0.0:8080	0.0.0.0:0	LISTENING
TCP	:::8080	:::0	LISTENING

이제 8080 포트에서 서버가 LISTENING 상태에 있는지 확인해 보겠습니다.
0.0.0.0:8080 또는 :::8080에서 IP를 지정해서 서버를 실행해 보겠습니다.
LISTENING 상태에 있는지 확인해 보겠습니다.

이제 서버가 LISTENING 상태에 있는지 확인해 보겠습니다. (이제 서버가 실행된 상태입니다.)

이제 서버가 LISTENING 상태에 있는지 확인해 보겠습니다. (이제 서버가 실행된 상태입니다.)

TCP	0.0.0.0:8080	0.0.0.0:0	ESTABLISHED
-----	--------------	-----------	-------------

8080 포트에서 서버가 LISTENING 상태에 있는지 확인해 보겠습니다.

[Linux #4] vi 和 nano



WSL 中 可以 编辑 系统 文件 (.bashrc, .vimrc, /etc/hosts, /etc/network/interfaces 等)

在 WSL 中 使用 vi 或 nano 编辑 系统 文件 的方法 如下。

使用 vi (或 vim)

打开 文件: 使用 vi, 或 vim

在 文件 中 编辑

保存 文件

i : 进入 插入 模式

a : 在 行 尾 插入

o : 在 行 尾 插入 新 行

Esc → 退出 插入 模式

Esc

! / 命令

:w : 保存

:q : 退出

:wq : 保存 并 退出

:q! : 强制退出

Ctrl

dd : 删除一行

yy : 复制一行

p : 粘贴

nano

Ctrl + O : 写入文件

Ctrl + /

Ctrl + O : 写入 (Write Out)

Enter : 换行

Ctrl + X : 退出

(Ctrl + O 写入文件)

Ctrl

Ctrl + K : 删除一行

Ctrl + U : 撤销

Ctrl + G : 退出

Ctrl

vi 模式

nano 文件

在文件前加 **sudo** 文件

sudo vi 文件

sudo nano 文件

文件

vi 在文件前加 **i** 进入插入模式 → 文件

nano 在文件前加 **Ctrl + O** → Enter → Ctrl + X

文件 文件

vi Esc → :wq

nano Ctrl + O → Enter → Ctrl + X

vi 在文件前加 **!!** 强制保存 "文件 文件 **!!**" 文件, 文件 文件 **Esc** 文件 **:q!**
文件 文件 文件 文件。

文件 文件 文件 文件 文件 文件 文件。

文件 文件:

```
sudo cp /etc/hosts /etc/hosts.bak
```

文件 .bak 文件 文件 文件 文件 文件。