

(참고) 기본적인 CLI 명령어

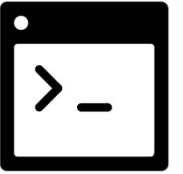
2021.01 Ver

CLI란?



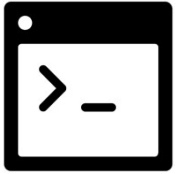
- Command Line Interface의 약자로 흔히 사용하는 GUI와 반대
- 클릭이 아닌 명령어를 이용하여 작업
- 윈도우에서는 기본적으로 CMD와 PowerShell을 제공하지만, 사용하는 명령어가 Mac이나 Linux 계열과는 차이가 있음.
- Git 강의에서 사용할 Git Bash는 Mac 및 Linux의 명령어를 일부 지원하기 때문에 해당 참고자료는 git bash 기준으로 작성

1. 프로세스 관련 명령어



- ps
- kill

ps



- 프로세스의 상태 보기
- **ps** : 프로세스의 상태 보기
- **ps aux** : CPU, 메모리 사용률, 프로세스 상태 코드 등 확인
- **ps -ef** : PID와 PPID 확인 가능, kill과 자주 쓰임

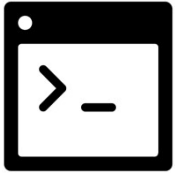
ps (Contd.)



```
jwseo@DESKTOP-O5QF9NF MINGW64 ~/cli-command-tutorial
$ ps -ef
```

UID	PID	PPID	TTY	STIME	COMMAND
jwseo	1677	1	?	09:27:31	/usr/bin/mintty
jwseo	1769	1678	pty0	09:57:31	/usr/bin/ps
jwseo	1678	1677	pty0	09:27:32	/usr/bin/bash

kill



- 프로세스 종료, ps와 자주 쓰임
- **kill (PID)** : 프로세스 종료
- **kill -9 (PID)** : 프로세스 강제 종료

kill (Contd.)



```
jwseo@DESKTOP-O5QF9NF MINGW64 ~/cli-command-tutorial
$ ps -ef
  UID      PID    PPID  TTY          STIME   COMMAND
  jwseo    1815    1678  pty0       10:00:15 /c/Users/jwseo/anaconda3/python
  jwseo    1677         1  ?          09:27:31 /usr/bin/mintty
  jwseo    1823    1678  pty0       10:00:22 /usr/bin/ps
  jwseo    1678    1677  pty0       09:27:32 /usr/bin/bash

jwseo@DESKTOP-O5QF9NF MINGW64 ~/cli-command-tutorial
$ kill -9 1815
[1]+  Exit 127                  python dummyProcess.py

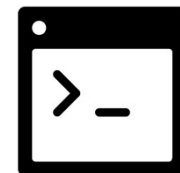
jwseo@DESKTOP-O5QF9NF MINGW64 ~/cli-command-tutorial
$ ps -ef
  UID      PID    PPID  TTY          STIME   COMMAND
  jwseo    1677         1  ?          09:27:31 /usr/bin/mintty
  jwseo    1830    1678  pty0       10:00:42 /usr/bin/ps
  jwseo    1678    1677  pty0       09:27:32 /usr/bin/bash
```

2. 파일 관련 명령어



- mkdir
- mv
- rm
- cp
- pwd
- cd
- touch
- ls
- find
- cat
- grep
- head
- tail
- vim

mkdir



- 새 디렉토리 생성
- **mkdir (directory)** : 새 directory 생성

```
jwseo@DESKTOP-O5QF9NF MINGW64 ~/cli-command-tutorial
$ ls -al
total 16
drwxr-xr-x 1 jwseo 197609 0 12월 31 10:03 ./
drwxr-xr-x 1 jwseo 197609 0 12월 31 09:59 ../

jwseo@DESKTOP-O5QF9NF MINGW64 ~/cli-command-tutorial
$ mkdir newDir

jwseo@DESKTOP-O5QF9NF MINGW64 ~/cli-command-tutorial
$ ls -al
total 16
drwxr-xr-x 1 jwseo 197609 0 12월 31 10:03 ./
drwxr-xr-x 1 jwseo 197609 0 12월 31 09:59 ../
drwxr-xr-x 1 jwseo 197609 0 12월 31 10:03 newDir/
```

mv



- 파일 또는 디렉토리 이동 및 이름 변경
- **mv (source file) (destination file)** : source의 이름을 destination으로 변경
- **mv (source file) (destination directory)** : source를 destination directory로 이동
- **mv (source file1) (source file2) (destination directory)** : source file1,2를 destination directory로 이동
- **mv (source directory) (destination directory)** : destination이 존재할 경우 source를 destination 하위로 이동, 그렇지 않으면 source의 이름을 destination으로 변경

mv (Contd.)



```
jwseo@DESKTOP-05QF9NF MINGW64 ~/cli-command-tutorial
$ ls -al
total 16
drwxr-xr-x 1 jwseo 197609 0 12월 31 10:04 ./
drwxr-xr-x 1 jwseo 197609 0 12월 31 09:59 ../
-rw-r--r-- 1 jwseo 197609 0 12월 31 10:04 file1
-rw-r--r-- 1 jwseo 197609 0 12월 31 10:04 file2
drwxr-xr-x 1 jwseo 197609 0 12월 31 10:03 newDir/

jwseo@DESKTOP-05QF9NF MINGW64 ~/cli-command-tutorial
$ mv file1 file2 newDir

jwseo@DESKTOP-05QF9NF MINGW64 ~/cli-command-tutorial
$ ls -al
total 16
drwxr-xr-x 1 jwseo 197609 0 12월 31 10:04 ./
drwxr-xr-x 1 jwseo 197609 0 12월 31 09:59 ../
drwxr-xr-x 1 jwseo 197609 0 12월 31 10:04 newDir/

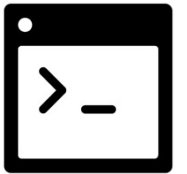
jwseo@DESKTOP-05QF9NF MINGW64 ~/cli-command-tutorial
$ ls -al newDir
total 0
drwxr-xr-x 1 jwseo 197609 0 12월 31 10:04 ./
drwxr-xr-x 1 jwseo 197609 0 12월 31 10:04 ../
-rw-r--r-- 1 jwseo 197609 0 12월 31 10:04 file1
-rw-r--r-- 1 jwseo 197609 0 12월 31 10:04 file2
```



rm

- 파일이나 디렉토리를 삭제하는 명령어
- **rm (file)** : 파일 삭제
- **rm -f (file)** : 파일 강제 삭제
- **rm -r (directory)** : 디렉토리 및 하위 항목 삭제
- **rm -rf (directory)** : 디렉토리 및 하위 항목 강제 삭제
- rm -rf는 진짜로 조심해서 사용!

rm (Contd.)



```
jwseo@DESKTOP-O5QF9NF MINGW64 ~/cli-command-tutorial
$ ls -al
total 16
drwxr-xr-x 1 jwseo 197609 0 12월 31 10:04 ./
drwxr-xr-x 1 jwseo 197609 0 12월 31 09:59 ../
drwxr-xr-x 1 jwseo 197609 0 12월 31 10:04 newDir/

jwseo@DESKTOP-O5QF9NF MINGW64 ~/cli-command-tutorial
$ rm -r newDir

jwseo@DESKTOP-O5QF9NF MINGW64 ~/cli-command-tutorial
$ ls -al
total 16
drwxr-xr-x 1 jwseo 197609 0 12월 31 10:05 ./
drwxr-xr-x 1 jwseo 197609 0 12월 31 09:59 ../
```

cp



- 복사하기
- **cp (source) (destination)** : source를 destination에 복사
- **cp -r (source) (destination)** : source를 하위 디렉토리 및 하위 파일까지 포함하여 destination에 복사
- **cp -a (source) (destination)** : source를 완전히 동일하게 destination에 복사.
수정시간, 소유자, 그룹, 퍼미션 등이 유지

cp (Contd.)



```
jwseo@DESKTOP-05QF9NF MINGW64 ~/cli-command-tutorial
$ ls -al
total 17
drwxr-xr-x 1 jwseo 197609 0 12월 31 10:08 ./
drwxr-xr-x 1 jwseo 197609 0 12월 31 10:07 ../
drwxr-xr-x 1 jwseo 197609 0 12월 31 10:08 destDir/
-rw-r--r-- 1 jwseo 197609 2 12월 31 10:07 file1

jwseo@DESKTOP-05QF9NF MINGW64 ~/cli-command-tutorial
$ cp file1 file2

jwseo@DESKTOP-05QF9NF MINGW64 ~/cli-command-tutorial
$ ls -al
total 18
drwxr-xr-x 1 jwseo 197609 0 12월 31 10:08 ./
drwxr-xr-x 1 jwseo 197609 0 12월 31 10:07 ../
drwxr-xr-x 1 jwseo 197609 0 12월 31 10:08 destDir/
-rw-r--r-- 1 jwseo 197609 2 12월 31 10:07 file1
-rw-r--r-- 1 jwseo 197609 2 12월 31 10:08 file2

jwseo@DESKTOP-05QF9NF MINGW64 ~/cli-command-tutorial
$ cp file1 destDir

jwseo@DESKTOP-05QF9NF MINGW64 ~/cli-command-tutorial
$ ls -al destDir
total 1
drwxr-xr-x 1 jwseo 197609 0 12월 31 10:09 ./
drwxr-xr-x 1 jwseo 197609 0 12월 31 10:08 ../
-rw-r--r-- 1 jwseo 197609 2 12월 31 10:09 file1
```

pwd



- 현재 디렉토리 출력
- **pwd** : 현재 디렉토리 출력

```
jwseo@DESKTOP-O5QF9NF MINGW64 ~/cli-command-tutorial  
$ pwd  
/c/Users/jwseo/cli-command-tutorial
```


cd



- 현재 디렉토리 이동
- **cd (directory)** : 해당 디렉토리로 이동
- (directory)가 비어 있으면 user directory로 이동

```
jwseo@DESKTOP-O5QF9NF MINGW64 ~/cli-command-tutorial
$ pwd
/c/Users/jwseo/cli-command-tutorial

jwseo@DESKTOP-O5QF9NF MINGW64 ~/cli-command-tutorial
$ cd destDir/

jwseo@DESKTOP-O5QF9NF MINGW64 ~/cli-command-tutorial/destDir
$ pwd
/c/Users/jwseo/cli-command-tutorial/destDir
```

touch



- 파일을 생성하거나 파일의 타임스탬프 갱신
- **touch (file)** : file이 없으면 빈 파일 생성하고, file이 있으면 타임스탬프를 갱신

```
jwseo@DESKTOP-05QF9NF MINGW64 ~/cli-command-tutorial
$ ls -al
total 18
drwxr-xr-x 1 jwseo 197609 0 12월 31 10:11 ./
drwxr-xr-x 1 jwseo 197609 0 12월 31 10:07 ../
-rw-r--r-- 1 jwseo 197609 2 12월 31 10:07 file1
-rw-r--r-- 1 jwseo 197609 2 12월 31 10:08 file2

jwseo@DESKTOP-05QF9NF MINGW64 ~/cli-command-tutorial
$ touch file3

jwseo@DESKTOP-05QF9NF MINGW64 ~/cli-command-tutorial
$ ls -al
total 18
drwxr-xr-x 1 jwseo 197609 0 12월 31 10:12 ./
drwxr-xr-x 1 jwseo 197609 0 12월 31 10:07 ../
-rw-r--r-- 1 jwseo 197609 2 12월 31 10:07 file1
-rw-r--r-- 1 jwseo 197609 2 12월 31 10:08 file2
-rw-r--r-- 1 jwseo 197609 0 12월 31 10:12 file3
```

ls



- 디렉토리 내용 목록 보기
- **ls (directory)** : 디렉토리의 목록 보기
- **ls -l (directory)** : 디렉토리의 목록 상세히 보기
- **ls -a (directory)** : 숨겨진 항목을 포함한 디렉토리 목록 보기
- **ls -al (directory)** : 숨겨진 항목을 포함한 디렉토리 목록 상세히 보기
- (directory) 위치가 비어 있을 경우 현재 디렉토리 대상
- 주로 directory를 지정하나, file이 올 경우 해당 파일의 정보 출력

ls (Contd.)



```
jwseo@DESKTOP-O5QF9NF MINGW64 ~/cli-command-tutorial
$ ls
file1 file2 file3

jwseo@DESKTOP-O5QF9NF MINGW64 ~/cli-command-tutorial
$ ls -l
total 2
-rw-r--r-- 1 jwseo 197609 2 12월 31 10:07 file1
-rw-r--r-- 1 jwseo 197609 2 12월 31 10:08 file2
-rw-r--r-- 1 jwseo 197609 0 12월 31 10:12 file3

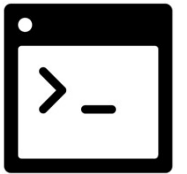
jwseo@DESKTOP-O5QF9NF MINGW64 ~/cli-command-tutorial
$ ls -al
total 18
drwxr-xr-x 1 jwseo 197609 0 12월 31 10:13 ./
drwxr-xr-x 1 jwseo 197609 0 12월 31 10:07 ../
-rw-r--r-- 1 jwseo 197609 0 12월 31 10:13 .hidden
-rw-r--r-- 1 jwseo 197609 2 12월 31 10:07 file1
-rw-r--r-- 1 jwseo 197609 2 12월 31 10:08 file2
-rw-r--r-- 1 jwseo 197609 0 12월 31 10:12 file3
```

find



- 파일 또는 디렉토리 검색 명령어
- **find (directory) -name (file)** : directory에서 file 찾기
- **find (directory) -type d** : directory에서 하위 디렉토리 전체 찾기
- **find (directory) -maxdepth (n) -type d** : directory에서 n단계 하위 디렉토리 찾기
- **find (directory) -type f** : directory에서 파일들 찾기

find (Contd.)



```
jwseo@DESKTOP-O5QF9NF MINGW64 ~/cli-command-tutorial
$ ls -al
total 16
drwxr-xr-x 1 jwseo 197609 0 12월 31 10:17 ./
drwxr-xr-x 1 jwseo 197609 0 12월 31 10:07 ../
-rw-r--r-- 1 jwseo 197609 0 12월 31 10:17 a
-rw-r--r-- 1 jwseo 197609 0 12월 31 10:17 b
-rw-r--r-- 1 jwseo 197609 0 12월 31 10:17 c

jwseo@DESKTOP-O5QF9NF MINGW64 ~/cli-command-tutorial
$ find . -name a
./a
```

cat



- 파일의 내용 출력
- **cat (file)** : 파일의 내용을 전부 출력, 후술할 head, tail, grep과 자주 쓰임

cat (Contd.)



```
jwseo@DESKTOP-05QF9NF MINGW64 ~/cli-command-tutorial
$ cat file
a
b
c
d
e
f
g
h
i
j
k
l
m
n
o
p
q
r
s
t
u
v
w
x
y
z
```




grep

- 패턴에 맞는 행들의 내용을 출력, 다른 명령어와 함께 쓰이는 경우가 대부분
- **grep (pattern) (file)** : file에서 pattern에 맞는 행 출력
- **ls (directory) | grep (pattern)** : ls의 출력 결과에서 pattern에 맞는 행 출력.
ls에 -l, -a, -al과 같은 옵션을 부여하여 사용 가능
- **cat (file) | grep (pattern)** : file의 내용 중 pattern과 일치하는 행 출력

```
jwseo@DESKTOP-O5QF9NF MINGW64 ~/cli-command-tutorial
$ ls -al | grep f
-rw-r--r-- 1 jwseo 197609 52 12월 31 10:19 file

jwseo@DESKTOP-O5QF9NF MINGW64 ~/cli-command-tutorial
$ cat file | grep f
f
```



head

- 처음 n개 행을 보여주는 명령어
- **head (file)** : file의 첫 10개행 출력
- **head -(n) (file)** : file의 첫 n개행 출력
- **cat (file) | head** : file의 첫 10개행 출력
- **cat (file) | head -(n)** : file의 첫 n개행 출력

```
jwseo@DESKTOP-O5QF9NF MINGW64 ~/cli-command-tutorial
$ cat file | head -3
a
b
c
```

tail



- 마지막 n개 행을 보여주는 명령어
- **tail (file)** : file의 끝 10개행 출력
- **tail -(n) (file)** : file의 끝 n개행 출력
- **cat (file) | tail** : file의 끝 10개행 출력
- **cat (file) | tail -(n)** : file의 끝 n개행 출력

```
jwseo@DESKTOP-O5QF9NF MINGW64 ~/cli-command-tutorial
$ cat file | tail -3
x
y
z
```

vim



- Linux계열에서 많이 쓰이는 텍스트 에디터
git bash에서도 파일을 편집 할 때 기본 에디터로 사용한다.
- **vim (file)** : file이 없으면 생성 후, vim에서 열기.

```
1 a
2 b
3 c
4 d
5 e
6 f
7 g
8 h
9 i
10 j
11 k
12 l
13 m
14 n
```

vim에서 file을 연 모습

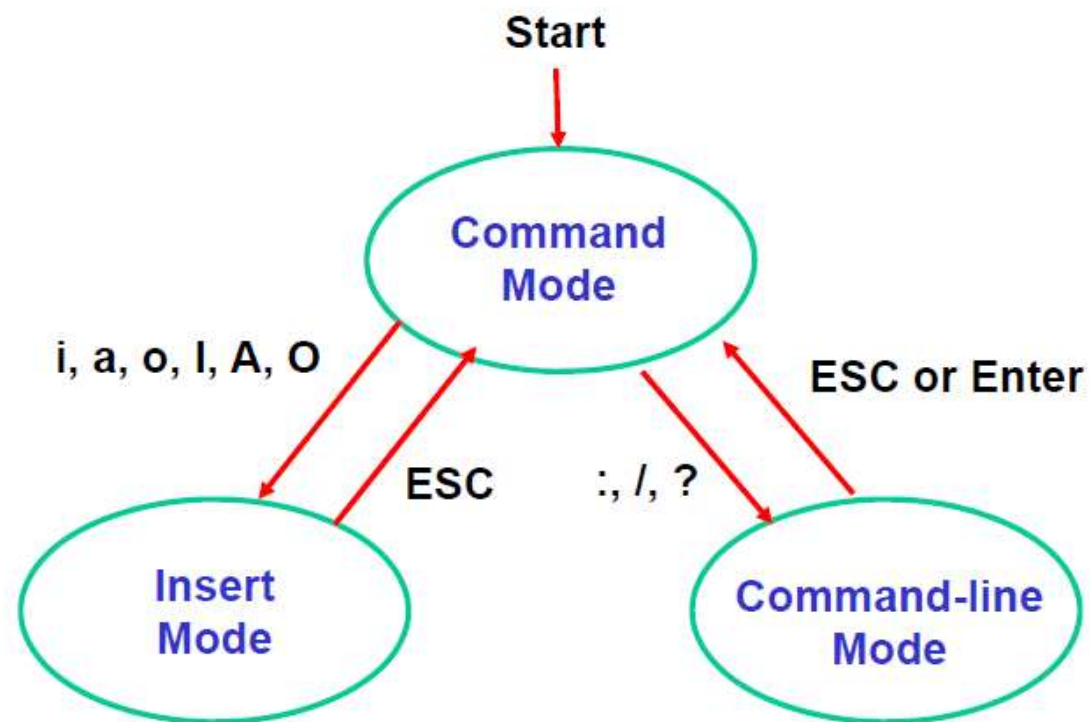
```
file [unix] (10:23 31/12/2020) 1,1 쪽 대 기
```

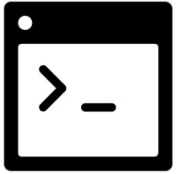
vim (Contd.)



- vim은 세가지 모드가 존재
 1. Insert Mode : 메모장처럼 파일을 수정 가능. **ESC**를 통해 comand mode로 이동 가능
 2. Command Mode : 명령을 통해 파일을 수정하거나 **i,a,o,I,A,O** 중 하나로 insert mode로 이동 가능하고, **:**를 통하여 command-line으로 이동 가능
 3. Command-line Mode : 명령어를 입력하여 파일을 저장하거나, 저장하지 않고 나가기, 파일에서 검색 등을 수행. **ESC**를 통하여 command mode로 이동 가능

vim (Contd.)





vim (Contd.)

사용순서는 다음과 같음

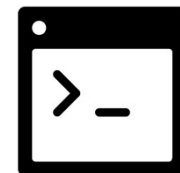
1. 터미널에서 **vim (file)**을 입력하여 vim을 실행한다.
2. **i,a,o,I,A,O** 중 하나를 입력하여 insert모드로 진입한다.
3. 파일을 편집한 후, esc를 눌러 command mode로 진입한다.
4. :를 통해 command-line mode로 진입한다.
5. 파일을 저장만 하려면 **w**를 입력, 저장하지 않고 나가려면 **q!**를 입력, 저장하고 나가려면 **wq**를 입력한다.



vim (Contd.)

- vim은 라인수를 보여준다거나, tab을 4로 지정하고, auto indent, syntax highlighting 기능을 사용하는 등 커스터마이징이 가능하다.
- **vim ~/.vimrc** : 커스터마이징 파일을 연다.
- 각 행에 명령어를 입력하면 하나의 옵션이 설정된다.

vim (Contd)



- 오른쪽 사진은 .vimrc의 예시
- **set nu**: 라인수 표시
- **set tabstop=(n)** : 탭 하나를 n칸으로 지정
- **set ai** : auto indent
(윗라인에 맞추어 들여쓰기)
- **set si** : smart indent
(함수나 조건문 등에서 들여쓰기)
- **syntax on** : 파일 확장자에 따라 syntax highlighting

```
set nu
set tabstop=4
set ai
set si
syntax on
```

3. 기타 명령어



- date

date



- 날짜 출력
- **date** : 날짜 출력

```
jwseo@DESKTOP-O5QF9NF MINGW64 ~/cli-command-tutorial  
$ date  
2020년 12월 31일 목 오 전 11:59:04
```

출처

제타위키 (<https://zetawiki.com>),
리눅스시스템프로그래밍 강의자료 (<http://mesl.khu.ac.kr>)