

# System Programming & OS 실습

## 4. Text editor: Vim

최민국

Dankook University

mgchoi@dankook.ac.kr

- Vim
- Vim - mode
- Vim install
- Vim 사용법
- Vim Plugin

- 
- The Vim logo, featuring a large, stylized 'V' with a green diamond shape behind it, and the word 'vim' in a smaller, italicized font below the 'V'.

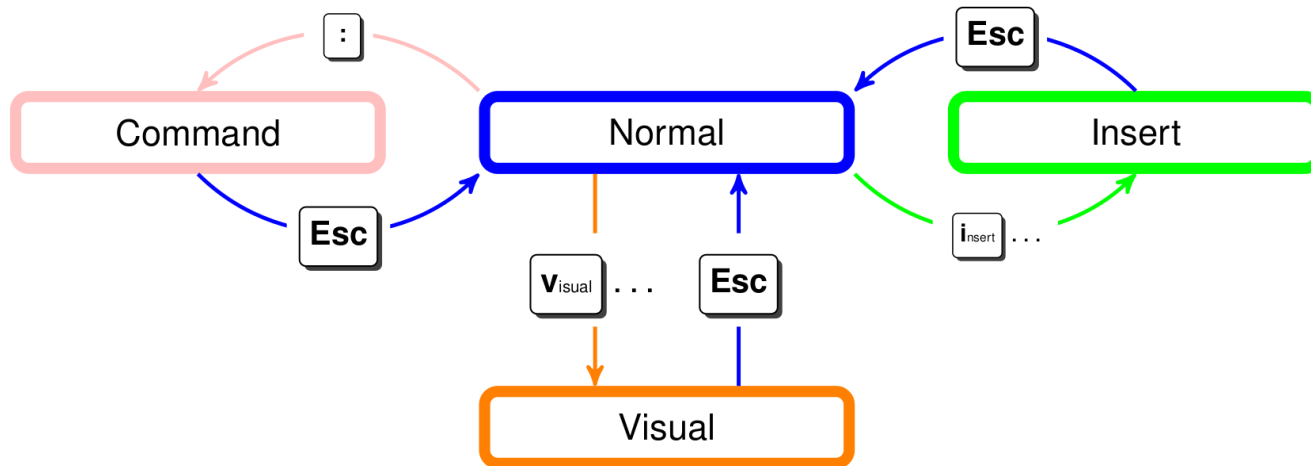


- Vim 편집기 명령어의 구조

<b>vim</b>	<b>[OPTION]</b>	<b>[ files... ]</b>
------------	-----------------	---------------------

- 자주 사용되는 옵션
  - -t TAG : TAG로 jump하여 에디터 시작
  - -o files... : 여러 파일을 split 모드로 열기
  - -R files... : read-only mode로 열기
  - -d files... : diff mode

- Vim에는 4가지의 모드가 존재한다.



- Normal
  - 처음 문서를 열었을 때의 모드
  - ESC를 통하여 normal 모드로 진입.
- Insert
  - 입력을 통해 해당 문자를 문서에 추가하는 모드
  - Normal 모드에서 i 입력을 통하여 진입
- Visual
  - 텍스트를 블록 단위로 조작 가능한 모드
  - Normal 모드에서 v 입력을 통하여 진입
- Command
  - 저장, 종료, 치환 등의 명령을 수행하는 모드
  - Normal 모드에서 : 입력을 통하여 진입

# Vim - install

- 02\_기본적인 리눅스 사용법에서 vim을 설치하였다.

```
[choi_gunhee@localhost ~]$ yum -y install vim
Error: This command has to be run with superuser privileges (under the root user on most systems).
[choi_gunhee@localhost ~]$
```

```
[choi_gunhee@localhost ~]$ sudo yum -y install vim

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

#1) Respect the privacy of others.
#2) Think before you type.
#3) With great power comes great responsibility.
```

```
[sudo] password for choi_gunhee: █
```

```
[sudo] password for choi_gunhee:
Last metadata expiration check: 0:21:28 ago on Sun 14 Aug 2022 08:34:54 PM KST.
.
Package vim-enhanced-2:8.2.2637-16.el9.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[choi_gunhee@localhost ~]$
```

- Vim의 실행, 저장, 종료

명령	기능
vim [파일명]	Vim 실행, 파일명을 지정하면 파일 열기
:q	종료 기능, 작업 내용이 없다면 바로 종료
:q!	작업한 내용을 저장하지 않고 종료
:w [파일명]	저장 기능, 파일명을 지정하면 새 파일로 저장
:wq, wq!	작업한 내용을 저장하고 종료
ZZ	작업한 내용을 저장하고 종료

- ```
[choi_gunhee@localhost ~]$ vim
```

```
[choi_gunhee@localhost ~]$ ls
Desktop    Downloads  Pictures   Templates  Videos
Documents  Music      Public     test       vim_testfile.txt
[choi_gunhee@localhost ~]$ cat vim_testfile.txt
Hello, Linux
[choi_gunhee@localhost ~]$ vim vim_testfile.txt
```

```

Hello, Linux

```

"vim testfile.txt" 1L, 13B 1,12 All



- Vim의 실행, 저장, 종료
  - Vim 종료
  - :q

```
VIM - Vi IMproved

      version 8.2.2637
    by Bram Moolenaar et al.
Modified by <bugzilla@redhat.com>
Vim is open source and freely distributable


  Become a registered Vim user!
type  :help register<Enter>  for information

type  :q<Enter>              to exit
type  :help<Enter> or <F1>   for on-line help
type  :help version8<Enter> for version info
```

```
:q
```

```
Hello, Linux
```

```
:q!
```

- Vim의 입력 모드
  - 진입 방법 : i 입력
  - ESC를 누르면 다시 normal 모드가 된다.

| 명령        | 기능                          |
|-----------|-----------------------------|
| i         | 현재 커서 앞에 입력                 |
| a         | 현재 커서 다음 자리에 입력             |
| o         | 커서가 위치한 행의 다음 행에 입력         |
| I (대문자 i) | 커서가 위치한 행의 첫 칼럼으로 이동 후 입력   |
| A         | 커서가 위치한 행의 마지막 컬럼으로 이동 후 입력 |
| O         | 커서가 위치한 행의 이전 행에 입력         |

- Normal mode에서 i를 입력하여 입력모드로 전환

```

Hello, Linux
~
~
[choi_gunhee@localhost ~]$ ls
Desktop    Downloads  Pictures  Templates  Videos      vim_testfile.txt
Documents  Music      Public    test        vim_hello.txt
[choi_gunhee@localhost ~]$
~
~

```

ESC를 누르면 normal mode 진입,  
:wq vim\_hello.txt 입력 후 enter를 누르면 저장 후 종료

```

~
~
:wq vim hello.txt

```

- Vim 실습 파일 만들기

**Hello, Linux**  
**I am Gunhee Choi**

**Seize the day.**  
**You only live once.**  
**This too shall pass away.**

**Thank you!**

위 내용을 작성하고,  
**vim\_hello.txt** 로 저장

```
Hello, Linux
I am Gunhee Choi

Seize the day.
You only live once.
This too shall pass away.
```

```
Thank you!
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
"vim_hello.txt" 8L, 104B
```

```
8,10
```

```
All
```

- Vim 실습 - 행 번호 보이기
  - Normal mode에서 :set nu 입력

```
Hello, Linux
I am Gunhee Choi

Seize the day.
You only live once.
This too shall pass away.
```

Thank you!

```
:set nu
```

```
1 Hello, Linux
2 I am Gunhee Choi
3
4 Seize the day.
5 You only live once.
6 This too shall pass away.
7
8 Thank you!
```

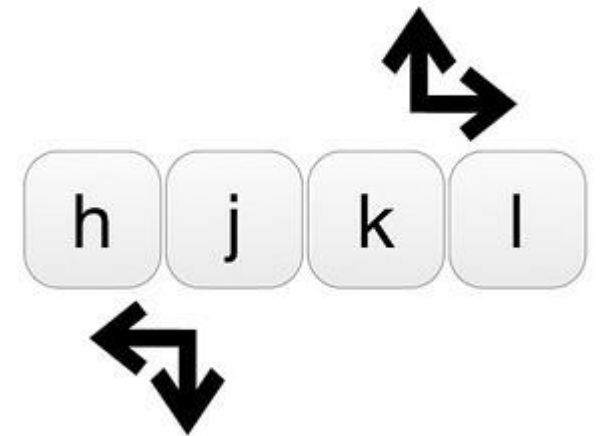
```
:set nu
```

8, 10

All

- Vim 실습 - 이동

| 명령          | 기능                       |
|-------------|--------------------------|
| k, ↑        | 커서를 한 행 위로 이동            |
| j, ↓        | 커서를 한 행 아래로 이동           |
| l(소문자 L), → | 커서를 한 문자 오른쪽으로 이동        |
| h, ←        | 커서를 한 문자 왼쪽으로 이동         |
| ^, 0(숫자)    | 커서를 현재 행의 처음으로 이동        |
| \$          | 커서를 현재 행의 마지막으로 이동       |
| -           | 커서를 이전 행의 처음으로 이동        |
| +, 엔터       | 커서를 다음 행의 처음으로 이동        |
| w           | 커서를 다음 단어의 첫 글자 위치로 이동   |
| b           | 커서를 앞 단어의 첫 글자 위치로 이동    |
| e           | 커서를 다음 단어의 마지막 글자 위치로 이동 |



## • Vim 실습 - 이동

| 명령          | 기능                       |
|-------------|--------------------------|
| k, ↑        | 커서를 한 행 위로 이동            |
| j, ↓        | 커서를 한 행 아래로 이동           |
| l(소문자 L), → | 커서를 한 문자 오른쪽으로 이동        |
| h, ←        | 커서를 한 문자 왼쪽으로 이동         |
| ^, 0(숫자)    | 커서를 현재 행의 처음으로 이동        |
| \$          | 커서를 현재 행의 마지막으로 이동       |
| -           | 커서를 이전 행의 처음으로 이동        |
| +, 엔터       | 커서를 다음 행의 처음으로 이동        |
| w           | 커서를 다음 단어의 첫 글자 위치로 이동   |
| b           | 커서를 앞 단어의 첫 글자 위치로 이동    |
| e           | 커서를 다음 단어의 마지막 글자 위치로 이동 |

```
1 Hello, Linux
2 I am Gunhee Choi
3
4 Seize the day.
5 You only live once.
6 This too shall pass away.
7
8 Thank you!
```

```
:set nu
```

```
8,10
```

```
All
```

- Vim 실습 - 화면 이동

| 명령       | 기능              |
|----------|-----------------|
| G        | 마지막 행으로 이동      |
| gg       | 처음 행으로 이동       |
| 숫자+G     | 지정한 행으로 이동      |
| :숫자      | 지정한 행으로 이동      |
| :\$      | 마지막 행으로 이동      |
| Ctrl + u | 반 화면 위로 이동      |
| Ctrl + d | 반 화면 아래로 이동     |
| Ctrl + b | 한 화면 위로 이동      |
| Ctrl + f | 한 화면 아래로 이동     |
| Ctrl + y | 화면을 한 행 위로 이동   |
| Ctrl + e | 화면을 한 행만 아래로 이동 |



- Vim 실습 - 화면 이동

| 명령       | 기능              |
|----------|-----------------|
| G        | 마지막 행으로 이동      |
| gg       | 처음 행으로 이동       |
| 숫자+G     | 지정한 행으로 이동      |
| :숫자      | 지정한 행으로 이동      |
| :\$      | 마지막 행으로 이동      |
| Ctrl + u | 반 화면 위로 이동      |
| Ctrl + d | 반 화면 아래로 이동     |
| Ctrl + b | 한 화면 위로 이동      |
| Ctrl + f | 한 화면 아래 이동      |
| Ctrl + y | 화면을 한 행 위로 이동   |
| Ctrl + e | 화면을 한 행만 아래로 이동 |

```
[choi_gunhee@localhost ~]$ cd test/
[choi_gunhee@localhost test]$ ls
adjtime_copy  adjtime_copy_mv  adjtime_ln  services_copy
[choi_gunhee@localhost test]$ vim services_copy
```

```
# /etc/services:
# $Id: services,v 1.49 2017/08/18 12:43:23 ovasik Exp $
#
# Network services, Internet style
# IANA services version: last updated 2016-07-08
#
# Note that it is presently the policy of IANA to assign a single well-known
# port number for both TCP and UDP; hence, most entries here have two entries
# even if the protocol doesn't support UDP operations.
# Updated from RFC 1700, ``Assigned Numbers'' (October 1994).  Not all ports
# are included, only the more common ones.
#
# The latest IANA port assignments can be gotten from
#   http://www.iana.org/assignments/port-numbers
# The Well Known Ports are those from 0 through 1023.
# The Registered Ports are those from 1024 through 49151
# The Dynamic and/or Private Ports are those from 49152 through 65535
#
# Each line describes one service, and is of the form:
#
# service-name port/protocol [aliases ...]  [# comment]
"services_copy" 11473L, 692252B 1,1 Top
```

- Vim 실습 - 편집 기능

| 명령        | 기능                                       |
|-----------|------------------------------------------|
| x, 숫자+x   | 커서가 위치한 문자 삭제, 숫자는 삭제할 문자 수              |
| dw, 숫자+dw | 커서가 위치한 단어를 삭제, 숫자는 삭제할 단어 수             |
| dd, 숫자+dd | 커서가 위치한 행 삭제, 숫자는 삭제할 행 수                |
| u         | 방금 수행한 명령 취소                             |
| U         | 한 행에 수행한 모든 명령 취소                        |
| :e!       | 마지막 저장한 내용 말고 전부 취소                      |
| yy,숫자+yy  | 커서가 위치한 행을 복사, 숫자는 복사할 행 수               |
| p         | 커서가 위치한 행의 아래쪽에 붙임                       |
| P         | 커서가 위치한 행의 위쪽에 붙임                        |
| dd, #dd   | 커서가 위치한 행 잘라내기, 숫자는 잘라낼 행의 수<br>삭제로 이용가능 |

- Vim 실습 - 편집 기능

| 명령        | 기능                                       |
|-----------|------------------------------------------|
| x, 숫자+x   | 커서가 위치한 문자 삭제, 숫자는 삭제할 문자 수              |
| dw, 숫자+dw | 커서가 위치한 단어를 삭제, 숫자는 삭제할 단어 수             |
| dd, 숫자+dd | 커서가 위치한 행 삭제, 숫자는 삭제할 행 수                |
| u         | 방금 수행한 명령 취소                             |
| U         | 한 행에 수행한 모든 명령 취소                        |
| :e!       | 마지막 저장한 내용 말고 전부 취소                      |
| yy,숫자+yy  | 커서가 위치한 행을 복사, 숫자는 복사할 행 수               |
| p         | 커서가 위치한 행의 아래쪽에 붙임                       |
| P         | 커서가 위치한 행의 위쪽에 붙임                        |
| dd, #dd   | 커서가 위치한 행 잘라내기, 숫자는 잘라낼 행의 수<br>삭제로 이용가능 |

```
1 Hello, Linux
2 I am Gunhee Choi
3
4 Seize the day.
5 You only live once.
6 This too shall pass away.
7
8 Thank you!
9 I am Gunhee Choi
10 I am Gunhee Choi
11 I am Gunhee Choi
12 I am Gunhee Choi
13 Thank you!
14 Thank you!
15 Thank you!
16 Thank you!
17 Hello, Linux
18 I am Gunhee Choi
19
20 Seize the day.
21 You only live once.
```

- Vim 실습 - 검색, 바꾸기 기능

| 명령                        | 기능                                                                  |
|---------------------------|---------------------------------------------------------------------|
| /string                   | string을 아래 방향으로 검색                                                  |
| ?string                   | string을 위 방향으로 검색                                                   |
| n                         | 원래 찾는 방향으로 다음 문자열 검색                                                |
| N                         | 역방향으로 다음 문자열 검색                                                     |
| :s/string1/string2        | 커서가 위치한 행의 string1을 string2로 바꿈                                     |
| :%s/string1/string2/g     | 파일 전체에서 모든 string1을 string2로 바꿈                                     |
| :<범위>s/string1/string2/   | <범위> 안의 모든 행에서 모든 string1을 string2로 바꿈                              |
| :<범위>s/string1/string2/g  | <범위> 안의 모든 행에서 모든 string1을 string2로 바꿈                              |
| :<범위>s/string1/string2/gc | <범위> 안의 모든 행에서 모든 string1을 string2로 바꿈<br>하지만, 바꾸기 전에 수정할지 여부를 묻는다. |

<범위> : 숫자, 숫자  
ex) 1,10

- Vim 실습 - 검색, 바꾸기 기능

```
1 Hello, Linux
2 I am Gunhee Choi
3
4 Seize the day.
5 You only live once.
6 This too shall pass away.
7
8 Thank you!
9 I am Gunhee Choi
10 I am Gunhee Choi
11 I am Gunhee Choi
12 I am Gunhee Choi
13 Thank you!
14 Thank you!
15 Thank you!
16 Thank you!
17 Hello, Linux
18 I am Gunhee Choi
19
20 Seize the day.
21 You only live once.
:/Gunhee
```

Gunhee 검색

```
1 Hello, Linux
2 I am Gunhee Choi
3
4 Seize the day.
5 You only live once.
6 This too shall pass away.
7
8 Thank you!
9 I am Gunhee Choi
10 I am Gunhee Choi
11 I am Gunhee Choi
12 I am Gunhee Choi
13 Thank you!
14 Thank you!
15 Thank you!
16 Thank you!
17 Hello, Linux
18 I am Gunhee Choi
19
20 Seize the day.
21 You only live once.
:%s/Thank/hi/g
```

Thank를 hi로 바꾸기

```
1 Hello, Linux
2 I am Gunhee Choi
3
4 Seize the day.
5 You only live once.
6 This too shall pass away.
7
8 hi you!
9 I am Gunhee Choi
10 I am Gunhee Choi
11 I am Gunhee Choi
12 I am Gunhee Choi
13 hi you!
14 hi you!
15 hi you!
16 hi you!
17 Hello, Linux
18 I am Gunhee Choi
19
20 Seize the day.
21 You only live once.
```

- Vim 실습 - Visual mode

| 명령       | 기능                      |
|----------|-------------------------|
| v        | 현재 커서 위치부터 블록 지정        |
| V        | 현재 커서의 위치에서 행 단위로 블록 지정 |
| Ctrl + v | 현재 커서의 위치에서 열 단위로 블록 지정 |
| y        | 블록 복사                   |
| d        | 블록 삭제                   |
| p        | 블록 붙여넣기                 |

```
1 Hello, Linux
2 I am Gunhee Choi
3
4 Seize the day.
5 You only live once.
6 This too shall pass away.
7
8 hi you!
9 I am Gunhee Choi
10 I am Gunhee Choi
11 I am Gunhee Choi
12 I am Gunhee Choi
13 hi you!
14 hi you!
15 hi you!
```

- Vim 실습 - 기타 명령어

| 명령              | 기능                          |
|-----------------|-----------------------------|
| :set nu         | 행 번호 표시                     |
| :set nonu       | 행 번호 숨기기                    |
| :set list       | 특수문자 표시                     |
| :set nolist     | 특수문자 숨기기                    |
| :set showmode   | 현재 모드 표시                    |
| :set noshowmode | 모드 표시를 숨기기                  |
| :set            | Set으로 설정한 모든 vim 변수 출력      |
| :set all        | 모든 vim 변수와 현재 값 출력          |
| ^(소문자 L)        | 화면 다시 출력                    |
| J               | 현재 행과 아래 행을 연결하여 하나의 행으로 만듦 |
| .               | 직전에 했던 명령 반복                |



## • CheatSheet

version 1.1  
April 1st, 06

### vi / vim 단축키 모음

Esc  
명령 모드

|             |               |             |             |             |              |            |                 |            |                 |           |           |           |
|-------------|---------------|-------------|-------------|-------------|--------------|------------|-----------------|------------|-----------------|-----------|-----------|-----------|
| ~ 대소문자 전환   | ! 외부 명령       | @ 매크로 실행    | # 이전 검색     | \$ 줄 끝으로 이동 | % 일치하는 괄호 찾기 | ^ 줄의 첫 글자  | & :s 반복         | * 다음 검색    | ( 문장 시작         | ) 문장 끝    | _ 아래줄로 이동 | + 다음 줄    |
| . 매크로 이동    | 1             | 2           | 3           | 4           | 5            | 6          | 7               | 8          | 9               | 0 줄의 처음   | - 이전 줄    | = 자동 들여쓰기 |
| Q 실행 모드     | W 다음 WORD     | E 끝 WORD    | R 수정        | T 뒤로 복사     | Y 줄단위 복사     | U 줄단위 실행취소 | I 시작에서 삽입       | O 행 위에 삽입  | P 커서 이전에 붙여넣기   | { 문단 시작   | }         | 문단 끝      |
| q 매크로 기록    | w 다음 단어       | e 단어 끝      | r 한 문자 교체   | t 한 문자 검색   | y 복사         | u 실행취소     | i 편집 모드         | o 행 아래에 삽입 | p 커서 이후에 붙여넣기   | [ 기타      | ]         | 기타        |
| A 줄 끝에 덧붙이기 | S 줄 삭제후 편집모드  | D 줄 끝까지 삭제  | F 뒤로 검색     | G 파일 끝으로 이동 | H 화면 상단      | J 줄 합치기    | K 도움말           | L 화면 하단    | : ex 명령줄        | " 레지스터 지정 | 열 이동      |           |
| a 덧붙이기      | s 단어 삭제후 편집모드 | d 1,3 삭제    | f 한 문자 찾기   | g 확장 명령     | h ←          | j ↓        | k ↑             | l →        | ; t/T/I/F 명령 반복 | ' 매크로 이동  | \ 사용 완료   |           |
| Z 종료        | X 백스페이스       | C 줄 끝까지 바꾸기 | V 줄단위 비주얼모드 | B 이전 WORD   | N 이전 (찾기)    | M 화면 가운데   | < 3 내어쓰기        | > 3 들여쓰기   | ? 3 찾기 (뒤로)     |           |           |           |
| Z 확장 명령     | X 글자 삭제       | c 1,3 바꾸기   | v 비주얼 모드    | b 이전 단어     | n 다음 (찾기)    | m 마크 설정    | , t/T/I/F 역순 검색 | . 명령 반복    | / 3 찾기          |           |           |           |

**동작** 커서를 이동하거나, 연산자가 동작할 범위를 지정합니다.

**명령** 바로 동작하는 명령, 빨간색은 편집 모드로 변경됩니다.

**연산자** 이동 관련 문자(숫자나 커서 이동)와 함께 사용하여 하며, 커서의 위치부터 목적지까지 연산합니다.

**확장** 특별한 키 합수로, 추가적인 키 입력이 필요합니다.

**q** 입력후 (숫자를 제외한 )으로 끝낼수 있는 글자를 입력하여야 합니다.

**words:** 구분자로 공백, 특수기호 모두 사용  
**WORDS:** 구분자로 공백 문자만 사용

**words:** quux(foo, bar, baz)  
**WORDS:** quux(foo, bar, baz)

**주요 명령행 명령 ('ex'):**  
:w (저장), :q (종료), :q! (저장하지 않고 종료)  
:e f (파일 f 열기),  
:s/s/x/y/g (파일 전체에서 'x'를 'y'로 교체),  
:h (vim 도움말), :new (새 파일)

**그외 중요한 명령들:**  
CTRL-R: 재실행 (vim),  
CTRL-F/-B: 페이지 위로/아래로,  
CTRL-E/-Y: 줄 스크롤 위로/아래로,  
CTRL-V: 블록비주얼 모드 (vim 전용)

**비주얼 모드:**  
커서를 움직여 지정한 범위에 연산자를 적용합니다. (vim 전용)

**참고:**

- (1) 복사/붙여넣기/지우기 명령어를 사용하기 전에 "x"를 입력하여 레지스터(클립보드)를 지정하세요. (x는 a에서 z 또는 \* 을 사용할 수 있음) (예: "ay\$를 입력하면 현재 커서에서 라인 끝까지의 내용을 레지스터 'a'에 저장합니다.)
- (2) 어떤 명령을 입력하기 전에 횟수를 지정하면, 횟수만큼 반복하게 됩니다.(예: 2p, d2w, 5l, d4j)
- (3) 연속으로 입력하는 명령은 현재의 라인에 반영됩니다. 예시: dd(현재 라인 지우기), >>(들여쓰기)
- (4) ZZ는 저장후 종료, ZQ는 저장하지 않고 종료.
- (5) zt: 커서가 위치한 곳을 제일위로 올리기, zb: 바닥으로, zz: 가운데로
- (6) gg: 파일의 처음으로(Vim 전용), gf: 커서가 위치한 곳의 파일 열기(Vim 전용)

vi/vim 에 대한 더 많은 강좌나 팁을 얻으려면 [www.viemu.com](http://www.viemu.com) (ViEmu, MS 비주얼 스튜디오를 위한 vi/vim 에뮬레이션)을 방문하십시오.



- Vim plugin들을 관리
- 온라인으로 목록을 불러와서 설치와 설정까지 가능

<https://github.com/VundleVim/Vundle.vim>

```

" Installing plugins to ~/home/starcraftman/.vim/bundle
Plugin 'gmarik/Vundle.vim'
Plugin 'Valloric/YouCompleteMe'
Plugin 'scrooloose/syntastic'
Plugin 'bling/vim-airline'
Plugin 'SirVer/ultisnips'
Plugin 'edsono/vim-matchit'
Plugin 'elzr/vim-json'
Plugin 'honza/vim-snippets'
Plugin 'justinmk/vim-sneak'
+ Plugin 'kien/ctrlp.vim'
Plugin 'ludovicchabant/vim-lawrencium'
Plugin 'majutsushi/tagbar'
Plugin 'mhinz/vim-signify'
+ Plugin 'plasticboy/vim-markdown'
Plugin 'scrooloose/nerdcommenter'
Plugin 'sjl/gundo.vim'
Plugin 'tpope/vim-fugitive'
Plugin 'tpope/vim-sleuth'
Plugin 'tpope/vim-surround'
Plugin 'tyru/open-browser.vim'
Plugin 'vim-scripts/a.vim'
Plugin 'tomasr/molokai'
> Plugin 'flazz/vim-colorschemes'
Helptags

~
~
~
~
~
~
~
~
~
~

" Color schemes
Plugin 'tomasr/molokai'
Plugin 'flazz/vim-colorschemes'

" Required, plugins available after.
call vundle#end()
filetype plugin indent on

<view> [Vundle] Installer 92% ↵ 24: 1 .vimrc 38% ↵ 322: 1
Processing 'flazz/vim-colorschemes'

```

- Vim Vundle

- Vundle git 페이지에서 복사하여 다음과 같은 명령어를 수행
- 위치 : home

```
git clone https://github.com/VundleVim/Vundle.vim.git ~/.vim/bundle/Vundle.vim
```

- Vim Vundle

- .vimrc 생성
- 위치 : home
- 옆의 내용을 복사 붙여넣기 필요  
(git에서 복사 할 것)

```
set nocompatible          " be iMproved, required
filetype off              " required

" set the runtime path to include Vundle and initialize
set rtp+=~/.vim/bundle/Vundle.vim
call vundle#begin()
" alternatively, pass a path where Vundle should install plugins
"call vundle#begin('~/.vim/path/plugins')

" let Vundle manage Vundle, required
Plugin 'VundleVim/Vundle.vim'

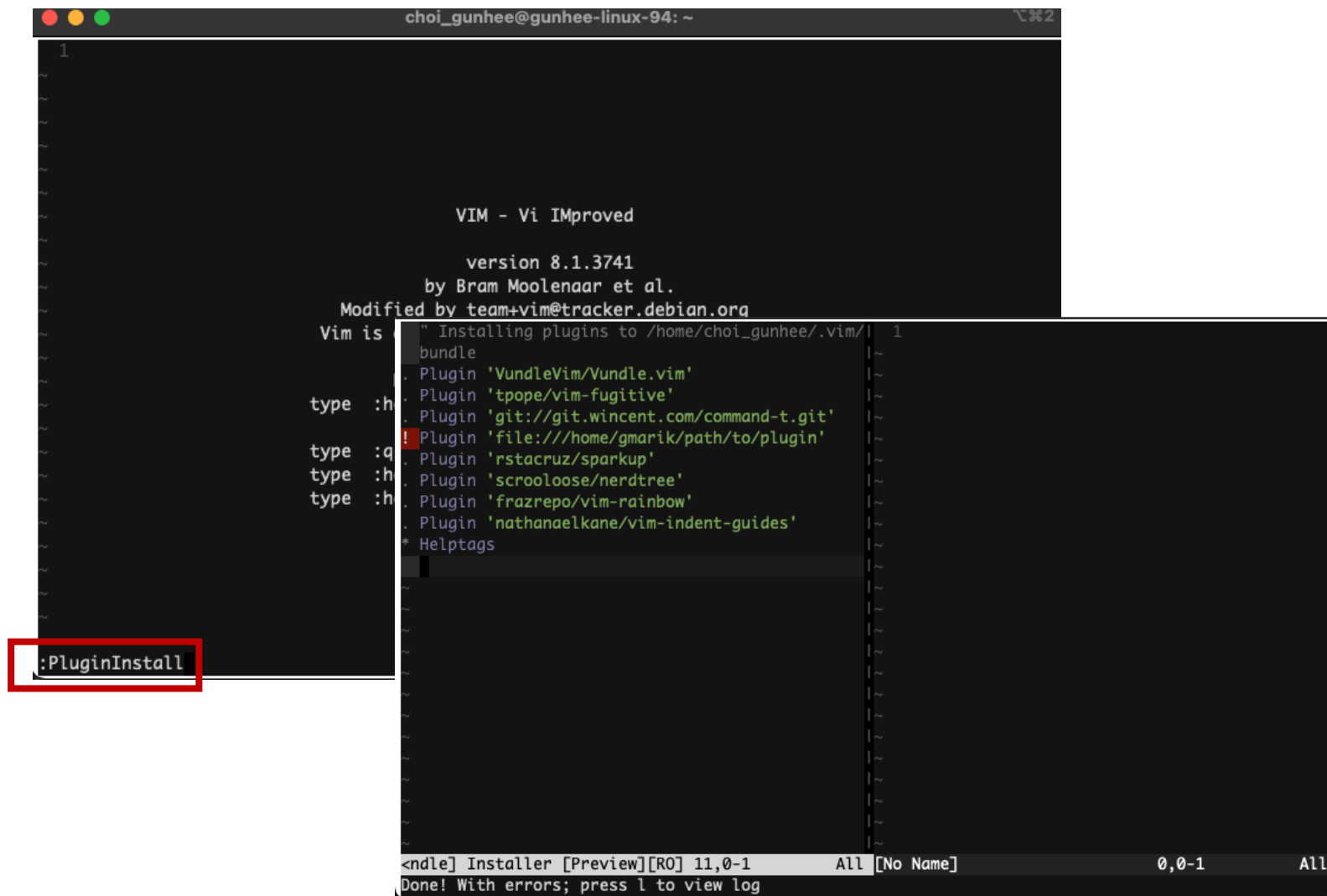
" The following are examples of different formats supported.
" Keep Plugin commands between vundle#begin/end.
" plugin on GitHub repo
Plugin 'tpope/vim-fugitive'
" plugin from http://vim-scripts.org/vim/scripts.html
" Plugin 'L9'
" Git plugin not hosted on GitHub
Plugin 'git://git.wincent.com/command-t.git'
" git repos on your local machine (i.e. when working on your own plugin)
Plugin 'file:///home/gmarik/path/to/plugin'
" The sparkup vim script is in a subdirectory of this repo called vim.
" Pass the path to set the runtimepath properly.
Plugin 'rstacruz/sparkup', {'rtp': 'vim/'}
" Install L9 and avoid a Naming conflict if you've already installed a
" different version somewhere else.
" Plugin 'ascenator/L9', {'name': 'newL9'}

" All of your Plugins must be added before the following line
call vundle#end()          " required
filetype plugin indent on  " required
" To ignore plugin indent changes, instead use:
"filetype plugin on
"
" Brief help
" :PluginList    - lists configured plugins
" :PluginInstall - installs plugins; append `!' to update or just :PluginUpdate
" :PluginSearch foo - searches for foo; append `!' to refresh local cache
" :PluginClean   - confirms removal of unused plugins; append `!' to auto-approve removal
"
" see :h vundle for more details or wiki for FAQ
" Put your non-Plugin stuff after this line
```

- Vim Vundle

- Vim 실행 후 명령모드에서  
:PluginInstall

- :q를 이용해서 종료



```
choi_gunhee@gunhee-linux-94: ~  
1  
  
VIM - Vi IMproved  
version 8.1.3741  
by Bram Moolenaar et al.  
Modified by team+vim@tracker.debian.org  
Vim is " Installing plugins to /home/choi_gunhee/.vim/ 1  
bundle  
Plugin 'VundleVim/Vundle.vim'  
Plugin 'tpope/vim-fugitive'  
Plugin 'git://git.wincent.com/command-t.git'  
! Plugin 'file:///home/gmarik/path/to/plugin'  
Plugin 'rstacruz/sparkup'  
Plugin 'scrooloose/nerdtree'  
Plugin 'frazrepo/vim-rainbow'  
Plugin 'nathanaelkane/vim-indent-guides'  
* Helptags  
type :h  
type :q  
type :h  
type :h  
:PluginInstall  
  
<ndle> Installer [Preview][R0] 11,0-1 All [No Name] 0,0-1 All  
Done! With errors; press l to view log
```

- .vimrc
  - 위치 : home
  - .vimrc 필요한 plugin 및 옵션 설정

```
drwxrwxr-x  4 choi_gunhee choi_gunhee
-rw-----  1 choi_gunhee choi_gunhee
-rw-rw-r--  1 choi_gunhee choi_gunhee
drwxrwxr-x  5 choi_gunhee choi_gunhee
-rw-rw-r--  1 choi_gunhee choi_gunhee
choi_gunhee@gunhee-linux-94:~$
```

```
4096  9월  11  00:56 .vim
13800 9월  12  09:43 .viminfo
1980  9월  11  01:22 .vimrc
4096  8월  29 14:02 .vscode-server
270   3월  13 2022 .wget-hsts
```

- .vimrc

- 위치 : home
- .vimrc 필요한 plugin 및 옵션 설정

```
42 " see :h vundle for more details or wiki for FAQ
43 " Put your non-Plugin stuff after this line
```

- .vimrc 파일 아래에는 위와 같은 문구 존재
- 이 문구 아래에 작성

- Nerdtree

- Plugin 'scroolouse/nerdtree'
- nmap <F9> :NERDTreeToggle<CR>

```
48 Plugin 'scroolouse/nerdtree'
49 nmap <F9> :NERDTreeToggle<CR>
50
```

- Ctrl+w+w

```
" Press ? for help  
..  
/home/choi_gunhee/test/  
test.c  
test2.txt  
~  
~  
~  
~  
~  
~
```

```
1 #include <stdio.h>  
2  
3 int main() {  
4     int total = 0;  
5  
6     printf("hello, world\n");  
7     for(int i=0; i<10; i++) {  
8         total += i;  
9     }  
10  
11     return 0;  
12 }
```

```
~  
~  
~
```

- Colors schem : jellybeans

<https://github.com/nanotech/jellybeans.vim>

- mkdir -p ~/.vim/colors
- cd ~/.vim/colors
- curl -O <https://raw.githubusercontent.com/nanotech/jellybeans.vim/master/colors/jellybeans.vim>
- 위 명령 실행



- Colors schem : jellybeans

<https://github.com/nanotech/jellybeans.vim>

colorscheme jellybeans

syntax on

```
45 colorscheme jellybeans
46 syntax on
```

위 옵션을 .vimrc에 입력

```
1 #include <stdio.h>
2
3 int main() {
4     int total = 0;
5
6     printf("hello, world\n");
7     for(int i=0; i<10; i++) {
8         total += i;
9     }
10
11     return 0;
12 }
```

```
1 #include <stdio.h>
2
3 int main() {
4     int total = 0;
5
6     printf("hello, world\n");
7     for(int i=0; i<10; i++) {
8         total += i;
9     }
10
11     return 0;
12 }
```

- Rainbow
  - Plugin 'frazrepo/vim-rainbow'
  - let g:rainbow\_active = 1

```
51 Plugin 'frazrepo/vim-rainbow'  
52 let g:rainbow_active = 1
```

<https://github.com/frazrepo/vim-rainbow>

```
rustc::usage(os::args()[0].clone())  
}  
fn find_cmd(command_string: &str) -> Option<Command> {  
    do COMMANDS.iter().find |command| {  
        command.cmd == command_string  
    }.map_move(|x| *x)  
}  
fn cmd_help(args: &[-str]) -> ValidUsage {  
    fn print_usage(command_string: -str) -> ValidUsage {  
        match find_cmd(command_string) {  
            Some(command) => {  
                match command.action {  
                    CallMain(prog, _) => println!(  
                        "The %s command is an alias for the %s program.",  
                        command.cmd, prog),  
                    => ()  
                }  
                match command.usage_full {  
                    UsgStr(msg) => println!("%s\n", msg),  
                    UsgCall(f) => f(),  
                }  
                Valid(0)  
            },  
            None => Invalid  
        }  
    }  
    match args {  
        [ref command_string] => print_usage((*command_string).clone()),  
        _ => Invalid  
    }  
}  
fn cmd_test(args: &[-str]) -> ValidUsage {  
    match args {  
        [ref filename] => {  
            let p = Path(*filename);  
            let test_exec = p.filestem().unwrap() + "test-";  
            invoke("rustc", &["--test", filename.to_owned(),  
                                ~"-o", test_exec.to_owned()], rustc::main_args);  
        }  
    }  
}
```

- Indent-guides
  - Plugin 'nathanaelkane/vim-indent-guides'
  - let g:indent\_guides\_enable\_on\_vim\_startup = 1

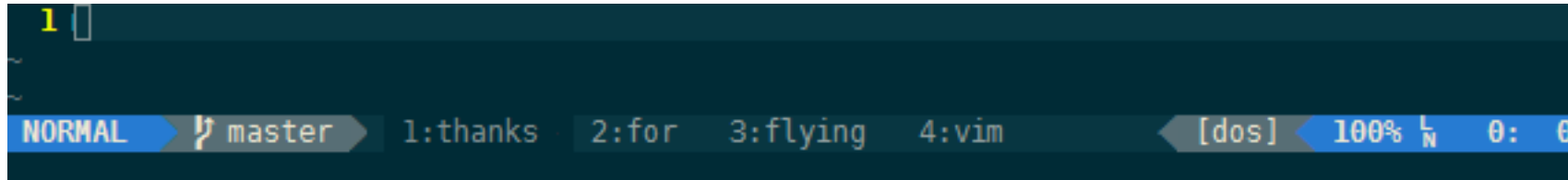
```
54 Plugin 'nathanaelkane/vim-indent-guides'  
55 let g:indent_guides_enable_on_vim_startup = 1
```

<https://github.com/nathanaelkane/vim-indent-guides>

```
40 # Enable user to choose a new password  
41 def lost_password  
42     redirect_to(home_url) && return unless Setting.lost_pass  
43     if params[:token]  
44         @token = Token.find_by_action_and_value("recovery", pa  
45         redirect_to(home_url) && return unless @token and !@to  
46         @user = @token.user  
47         if request.post?  
48             @user.password, @user.password_confirmation = params  
49             if @user.save  
50                 @token.destroy  
51                 flash[:notice] = l(:notice_account_password_update  
52                 redirect_to :action => 'login'  
53                 return  
54             end  
55         end  
56         render :template => "account/password_recovery"  
57         return  
58     else  
59         if request.post?  
60             user = User.find_by_mail(params[:mail])  
61             # user not found in db  
62         end  
63     end  
64 end  
65  
66 colorscheme bclear  
67 set ts=2 sw=2 et  
68 let g:indent_guides_start_level = 2
```

- VIM-AIRLINE
  - Plugin 'vim-airline/vim-airline'

<https://github.com/vim-airline/vim-airline>



- Vim 옵션
  - syntax on : 문법 강조
  - set hlsearch : 검색 매칭 문자열 강조
  - set nu : 줄 번호 표시
  - set autoindent : 자동 들여쓰기
  - set cindent : 자동 들여쓰기 (c언어용)
  - set ts=4 : '/t'에서 몇 개의 스페이스로 할 것인지
  - set sts=4 : tab을 눌렀을 때, 몇 개의 스페이스로 할 것인지
  - set shiftwidth=4 : 자동 들여쓰기 할 때, 몇개의 스페이스로 할 것 인지

- Vim 옵션
  - set laststatus=2 : statusline을 보여주는 값  
(0 :출력안함, 1:창 2개이상 일때, 2:항상)
  - set showmatch : 닫는 괄호 입력 시, 짝이 되는 괄호를 하이라이트
  - set ruler : 커서 위치의 줄번호와 행번호 출력
  - set fileencodings=utf8,euc-kr : 인코딩 형식 지정

- Vim 옵션

- Smart setting : 검색, 탭, 들여쓰기를 자동으로 설정
  - set smartcase
  - set smarttab
  - set smartindent
- set ignorecase : 검색 시 대소문자 무시
- set incsearch : 단어 검색시 글자 입력할 때마다 검색

- Vim 창 분할
  - sp : 가로 창 분할
  - vp : 세로 창 분할
- 이동
  - Crtl + w, w : 다음 창으로 커서
  - Crtl + w, W : 이전 창으로 커서 이동
  - Crtl + w, [H, J, K, L] : 해당 방향으로 커서 이동, vim 에서는 H, J, K, L이 방향키를 대체 가능
- Crtl + w, r : 다음 화면과 위치 바꾸기
- 종료 : q, 모든 창 종료 : qa



- ctags
  - 소스 파일 내에 정의된 전역 변수, 함수, 매크로, 구조체의 정보를 데이터베이스 파일로 생성하는 툴
- 설치
  - `yum install ctags`
- 데이터베이스 생성
  - `ctags -R`
- 경로 설정
  - 위치 : `vimrc`
  - `set tags=/home/user/workspace/project/tags`
  - `set tags=./tags, /usr/src/linux/tags -> 콤마로 여러 개 가능`

- ctags

- 소스 파일 내에 정의된 전역 변수, 함수, 매크로, 구조체의 정보를 데이터베이스 파일로 생성하는 툴

```
choi_gunhee@gunhee-linux-94:~/MyProject/Linux_Kernel_study/linux-5.16.13$ ls
arch      CREDITS      Documentation  init      kernel      Makefile     samples     tools
block     crypto       drivers        ipc       lib          mm           scripts     usr
certs     cscope.files fs            Kbuild    LICENSES    net          security    virt
COPYING   cscope.out   include        Kconfig   MAINTAINERS README       sound
choi_gunhee@gunhee-linux-94:~/MyProject/Linux_Kernel_study/linux-5.16.13$ ctags -R
choi_gunhee@gunhee-linux-94:~/MyProject/Linux_Kernel_study/linux-5.16.13$ ls
arch      CREDITS      Documentation  init      kernel      Makefile     samples     tags
block     crypto       drivers        ipc       lib          mm           scripts     tools
certs     cscope.files fs            Kbuild    LICENSES    net          security    usr
COPYING   cscope.out   include        Kconfig   MAINTAINERS README       sound       virt
choi_gunhee@gunhee-linux-94:~/MyProject/Linux_Kernel_study/linux-5.16.13$
```

- ctags
  - 소스 파일 내에 정의된 전역 변수, 함수, 매크로, 구조체의 정보를 데이터베이스 파일로 생성하는 툴

```
set tags=/home/choi_gunhee/MyProject/Linux_Kernel_study/linux-5.16.13/tags
```

- set tags=경로/tags
- 위 옵션을 .vimrc에 입력

- ctags
  - 소스 파일 내에 정의된 전역 변수, 함수, 매크로, 구조체의 정보를 데이터베이스 파일로 생성하는 툴
  - Ctrl + ] : 함수나 구조체 정의로 이동
  - Ctrl + t : 이전 위치로 이동

- cscope
  - ctags로 찾기 힘든 전역 변수나 함수의 호출 등을 볼 때 사용
- 설치
  - `yum install cscope`
- 쉘 스크립트 생성 (mkcscope.sh)
  - `#!/bin/bash`
  - `rm -rf cscope.files cscope.out`
  - `find . `pwd` \( -name '*.c' -o -name '*.cpp' -o -name '*.cc' -o -name '*.h' -o -name '*.s' -o -name '*S' \) -print > cscope.files cscope -i cscope.files`

- cscope

- ctags로 찾기 힘든 전역 변수나 함수의 호출 등을 볼 때 사용

- `chmod 755 mkcscope.sh`

- `mv mkcscope.sh /usr/local/bin`

- 언제나 실행하기 위해서 bin으로 이동

```
choi_gunhee@gunhee-linux-94:~/MyProject/Linux_Kernel_study/linux-5.16.13$ ls
arch      CREDITS   Documentation  init      kernel      Makefile  samples  tags
block     crypto    drivers        ipc       lib          mm        scripts  tools
certs     cscope.files  fs          Kbuild   LICENSES    net       security  usr
COPYING   cscope.out  include       Kconfig  MAINTAINERS  README    sound     virt
choi_gunhee@gunhee-linux-94:~/MyProject/Linux_Kernel_study/linux-5.16.13$
```

- 프로젝트 최상위에서 스크립트 실행

- `mkcscope.sh cscope.files cscope.out`
- 생성이 완료되면, Ctrl + d 입력 후 종료

- cscope
  - ctags로 찾기 힘든 전역 변수나 함수의 호출 등을 볼 때 사용
- vimrc에 다음과 같은 내용 입력
  - set csprg=/usr/bin/cscope
  - set csto=0
  - set cst
  - set nocsverb
  - if filereadable("./cscope.out")
  - cs add cscope.out
  - else
  - cs add /usr/user/workspace/project/cscope.out (프로젝트경로)
  - endif
  - set csverb

- cscope
  - ctags로 찾기 힘든 전역 변수나 함수의 호출 등을 볼 때 사용
- 사용법
  - cs find [유형] 검색어
  - Ex) cs find 1 inode

|        |                       |
|--------|-----------------------|
| 0 또는 s | C 심볼 검색               |
| 1 또는 g | 정의 검색                 |
| 2 또는 d | 호출되는 함수 검색            |
| 3 또는 c | 호출하는 함수 검색            |
| 4 또는 t | 텍스트 문자열 검색            |
| 6 또는 e | 확장 정규식을 이용한 검색        |
| 7 또는 f | 파일 이름 검색              |
| 8 또는 i | 본 파일을 include하는 파일 검색 |