리눅스 시스템 관리

3.1 텍스트 에디터

- 텍스트 에디터는 문자 기반으로 파일을 생성하고 수정할 수 있는 프로그램
- 프로그래밍에서는 소스 코드의 입력과 수정이 가장 필수적인 작업니다.
- 유닉스에서는 다음과 같은 에디터를 지원하고 있다.

ed/ex
메모리량이 작고 속도는 빠르지만 라인단위로 편집하기 때문에 조금 불편하다

Vi(vim)
대부분 유닉스에서 지원하는 에디터로 강력한 기능을 제공하지만 기능이 너무 많은 것이 단점이기도 하다.

GNU nano
유닉스에서 지원하는 에디터로 vi보다 편리한 편집 기능을 제공한다.

Emacs
GNU를 만든 리처드 스톨만이 개발한 에디터 이다. 유닉스에서는 기본적으로 제공하지 않기때문에 별도의 설치가 필요하다.

- Vi 에디터는 BSD 유닉스용으로 최초에 개발되었지만 거의 모든 유닉스에서 기본적으로 제공하고 있으며 리눅스에서는 보다 편하게 사용하도록 성능을 개선한 vim(iMproved)을 사용한다.

3.2.1 vi 에디터 시작하기

```
$ vi 파일명
```

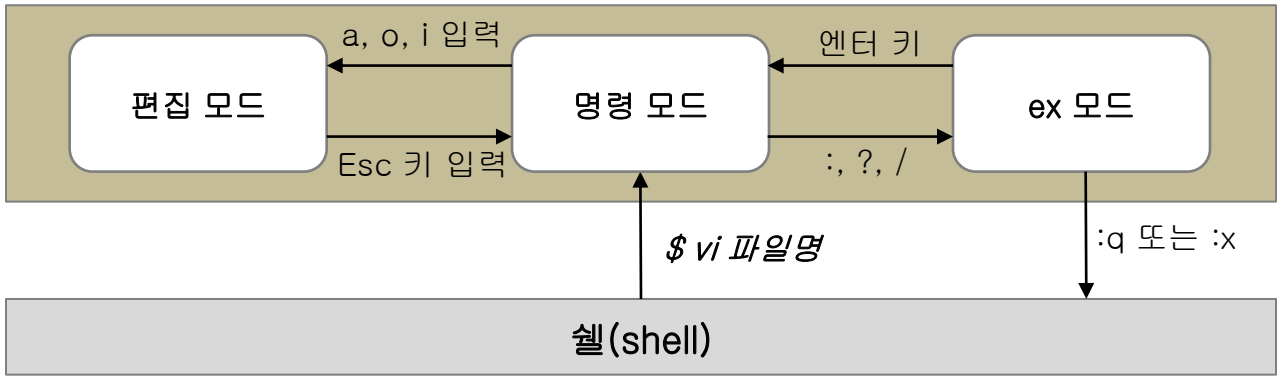
지정한 파일명이 있으면 새 파일을 열고 있으면 기존의 파일을 열게 된다.

3.2.2 vi 에디터 모드

명령(Command) 모드
명령을 수행할 수 있는 모드로 파일의 내용을 입력할 수 없다.

편집(Edit) 모드
문자 입력과 수정을 할 수 있는 모드

ex 모드
화면의 제일 아래 행에 명령어를 입력할 수 있는 모드



Vi 에디터

3.2.3 입력/편집 모드

다음 명령키로 명령 모드에서 입력/편집 모드로 문서 편집작업을 시작할 수 있다.

명령키	수행 작업
i	커서 앞에 삽입
a	커서 뒤에 삽입
o	현재 줄 다음에 삽입
I	현재 줄 첫 칸 앞에 텍스트 입력
A	현재 줄 끝에 텍스트 입력
O	현재 줄 앞에 삽입

3.2.4 종료

저장명령
저장하거나 종료하려면 “명령모드”로 돌아와야 함

명령키	수행 작업
:w ↵	현재의 파일명으로 파일 저장
:w 파일명 ↵	지정한 파일명으로 파일 저장

종료명령
저장 후 종료 또는 그냥 종료

명령키	수행 작업
:q ↵	작업 내용을 저장하였으면 vi 종료
:q! ↵	작업내용을 저장하지 않고 vi 종료
:wq↵	작업 내용을 저장한 후 vi 종료
:wq 파일명↵	작업 내용을 지정한 파일명으로 저장한 후 vi 종료
zz (shift-zz)↵	작업 내용을 저장한 후 vi 종료

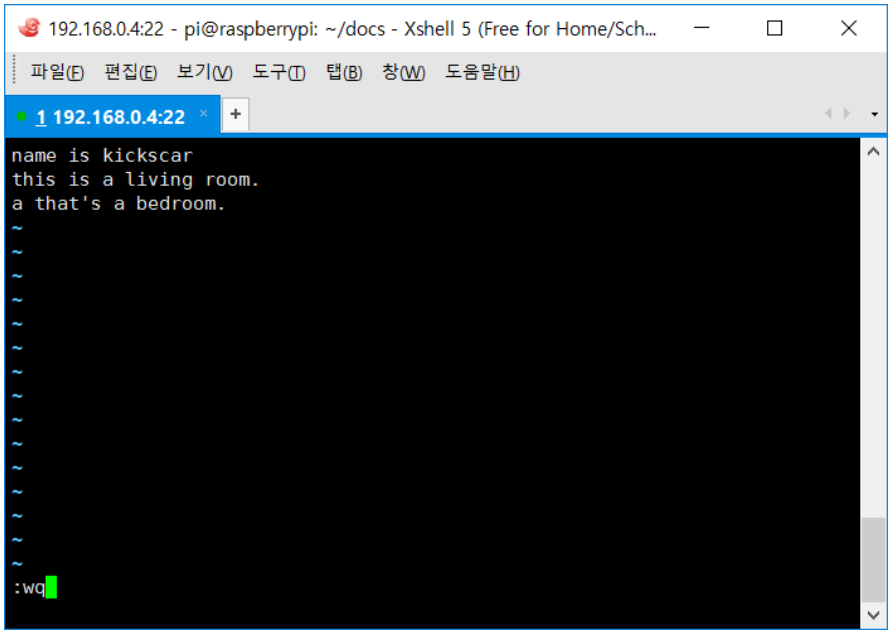
3.2 vi 에디터

[실습]

1. 실습 디렉토리 만들기

$ mkdir docs
$ cd docs
$ vi test.txt

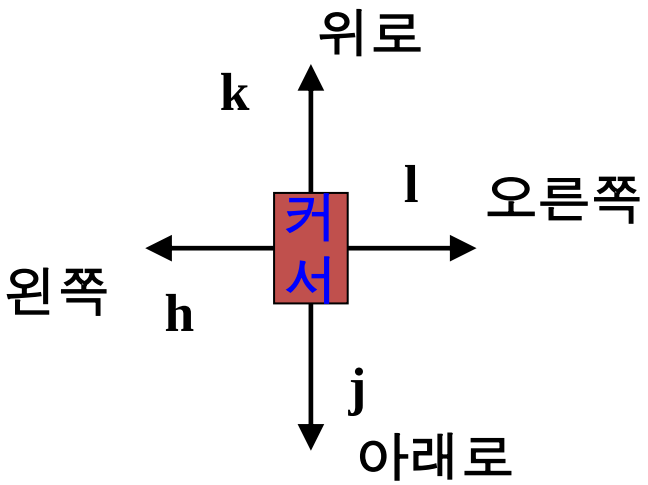
2. 문서 편집



작성 중 틀릴 경우 ESC -> :q! (저장안하고 빠져 나옴) -> *$ vi test.txt*
위의 세 줄을 정확하게 작성 한 후 ESC -> :wq 로 정상적으로 저장하고 종료 합니다.

3.2.4 커서의 이동

- 편집 모드에서는 커서를 움직일 수 없다.
- 편집 모드에서 커서를 움직이기 위해 ESC키를 눌러 명령 모드에서 다음 키로 커서를 움직인다.
- **화살표로 이동 가능(vim)**
- h, j , k, l 키로 이동



이동	명령어
한 행 위	k
한 행 아래	j
한 문자 오른 쪽	l
한 문자 왼쪽	h
행의 시작	^ 또는 0
행의 마지막	$
이전 행의 처음	-
다음 행의 처음	+ 또는 ↵

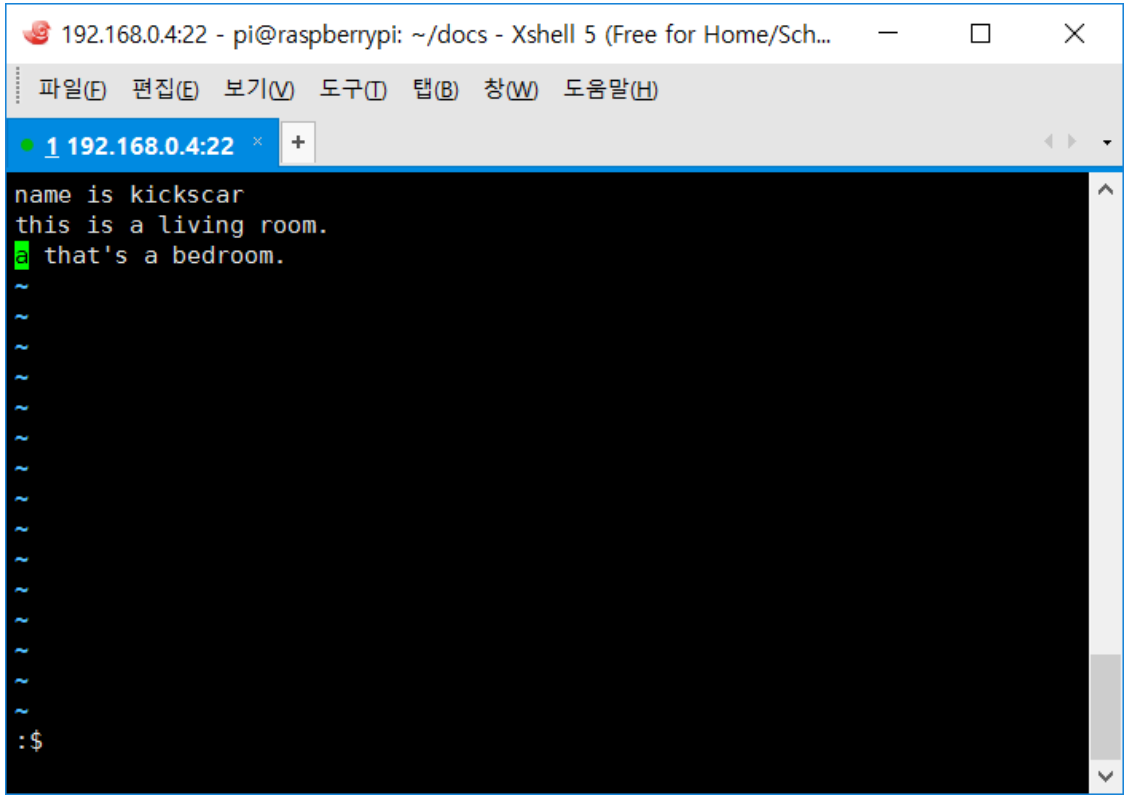
3.2.4 커서의 이동

- 지정한 곳으로 이동

이동	명령키
줄 번호 n 위치로	:n 또는 nG
파일의 끝 줄로 이동	:$ 또는 G
n줄 만큼 앞으로 이동	n+
n줄 만큼 뒤로	n-
현재 문장의 처음으로	(
다음 문장의 처음으로	)
현재 문단의 처음으로	{
다음 문단의 처음으로	}

3.2 vi 에디터

[실습]



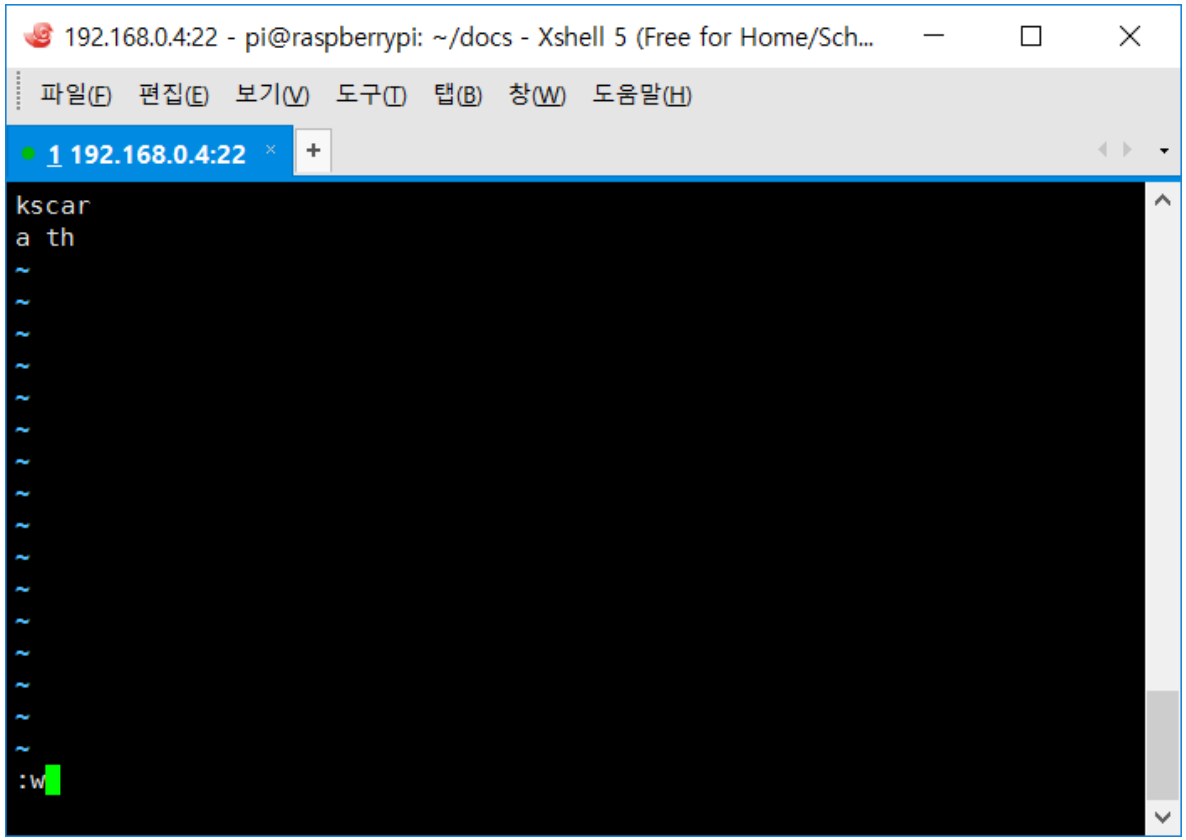
1) 커서를 1행으로 이동 : 1G 또는 :1
2) 1행의 두 번째 단어로 이동 : w
3) 2행으로 이동 : j
4) 커서를 좌로 이동 : l
5) 마지막행으로 이동 : G 또는 :$

3.2.5 삭제 및 취소

명령 모드에서 동작

명령어	삭제 대상	수행 작업
x, #x	문자	커서 위치의 문자 삭제(예:3x)
dw, #dw	단어	커서 위치의 단어 삭제
dd, #dd	줄	커서 위치의 줄 삭제
D(shift-d)	줄의 일부	커서 위치부터 줄 끝까지 삭제
u		방금 수행한 명령 취소
U		해당 줄의 모든 편집 취소

[실습]



1) 1G
2) 11x
3) jj
4) llll
5) D
6) k
7) dd
8) u
9) dd
10) :w

3.2.6 편집 기능 - 복사 / 잘라내기 / 붙이기

명령 모드에서 동작

명령어	수행 작업
yy, #yy	현재 행을 버퍼로 복사 (예:4yy)
p	현재 행 다음에 버퍼 내용 삽입
P	현재 행 위쪽에 버퍼 내용을 삽입
dd, #dd	현재 행을 잘라내기

버퍼

vi는 작업 내용을 버퍼에 저장 – 실행 취소 가능
복사하기, 잘라내기에 사용

버퍼종류

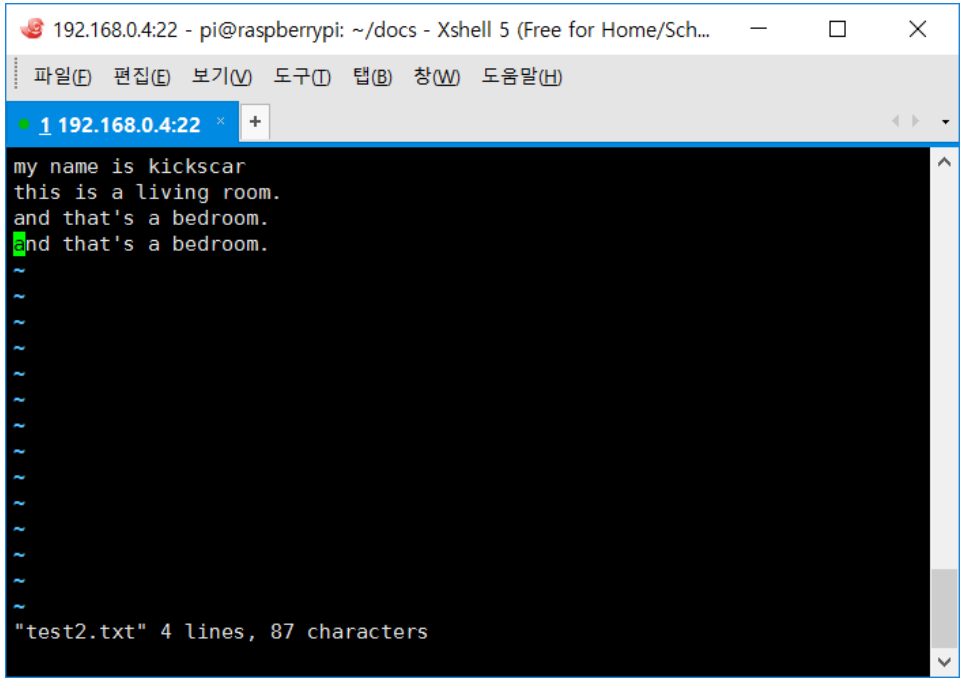
Unnamed buffer (이름 없는 버퍼)
Named buffers (이름이 있는 버퍼) “a, “b, … “z
Numbered buffers(번호가 있는 버퍼) “1, “2, …,“9

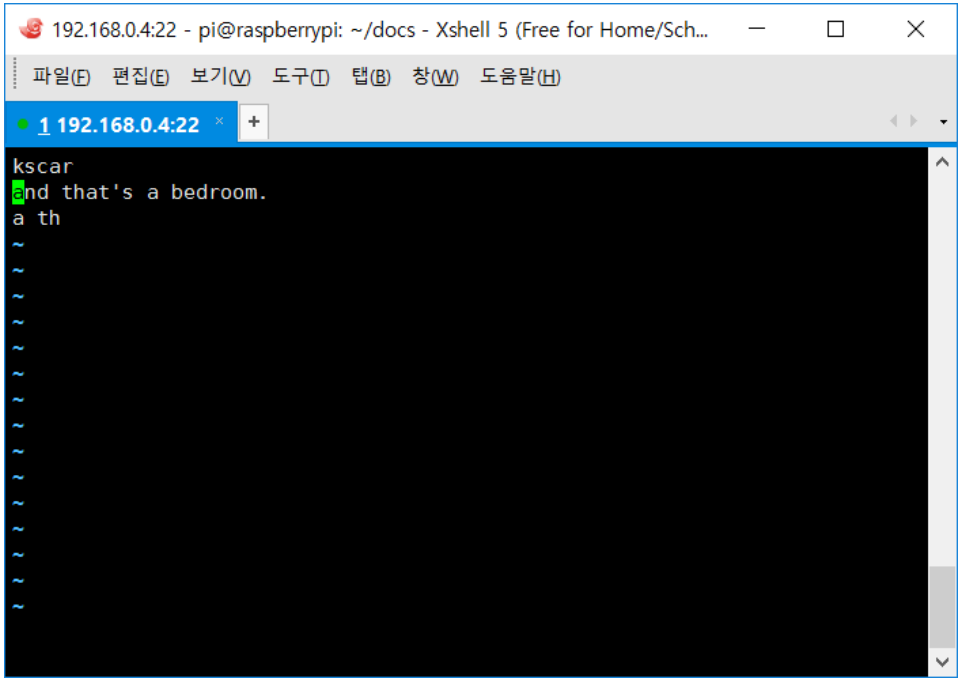
사용 예

“a3yy -> 현재 행부터 아래로 3줄을 a버퍼에 저장
“ap -> a버퍼의 내용을 붙이기

3.2 vi 에디터

[실습]





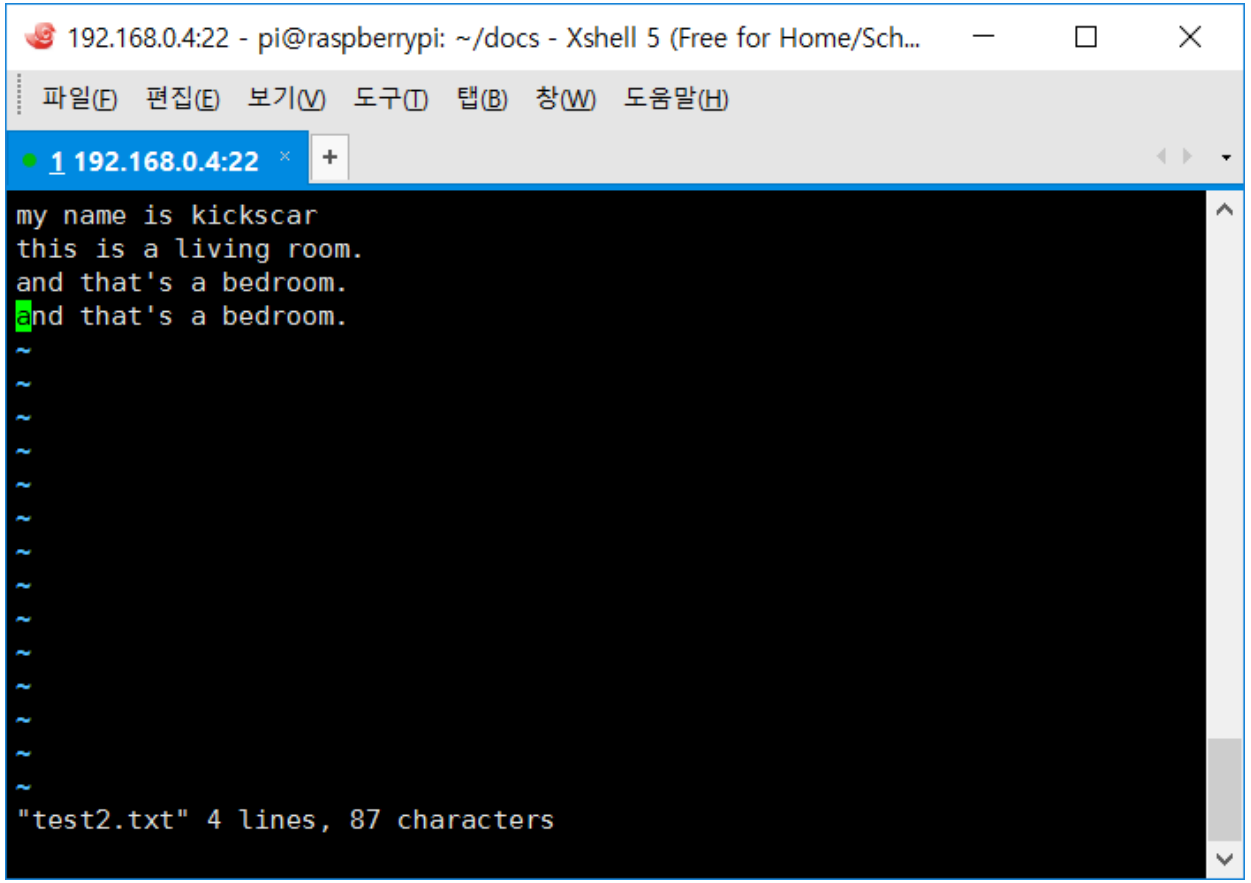
1) :3 2) yy 3) p 4) k 5) dd 6) p 7) “ayy 8) :w 9) :e test.txt 10)“ap

3.2.7 검색 기능

ex 모드에서 동작

명령어	수행 작업
/문자열	현재 위치부터 파일 앞쪽으로 문자열 탐색
?문자열	현재 위치부터 파일 뒤쪽으로 문자열 탐색
n	다음 문자열 탐색
N	역방향으로 문자열 탐색

[실습]



1) /room
2) n
3) ?name
4) n
5) N

3.2.8 기타 기능

vi에서 쉘 명령 실행

명령어	수행 작업
:!명령	vi를 중단하고 지정한 명령 수행 (vi로 돌아올 때 : ↵)
:sh	vi를 잠시 빠져나가서 쉘을 수행 (vi로 돌아올때 : exit)

3.2.8 기타 기능

알아두면 유용한 명령 키

명령어	수 행
:f 파일명	파일 이름을 지정한 이름으로 변경
:w %.old	현재 파일을 .old 이름으로 저장해 둘 때
^g	기본적인 파일정보 출력(파일명, 라인 수 등)
J	현재 줄과 다음 줄 연결
.	바로 이전에 수행한 명령 재 실행
~	현재 커서 위치의 한 문자를 소문자 혹은 대문자로 전환