

Chapter 03 문서 편집

목차

00 개요

01 리눅스의 문서 편집기

02 vi 사용법

03 vi 환경 설정

학습목표



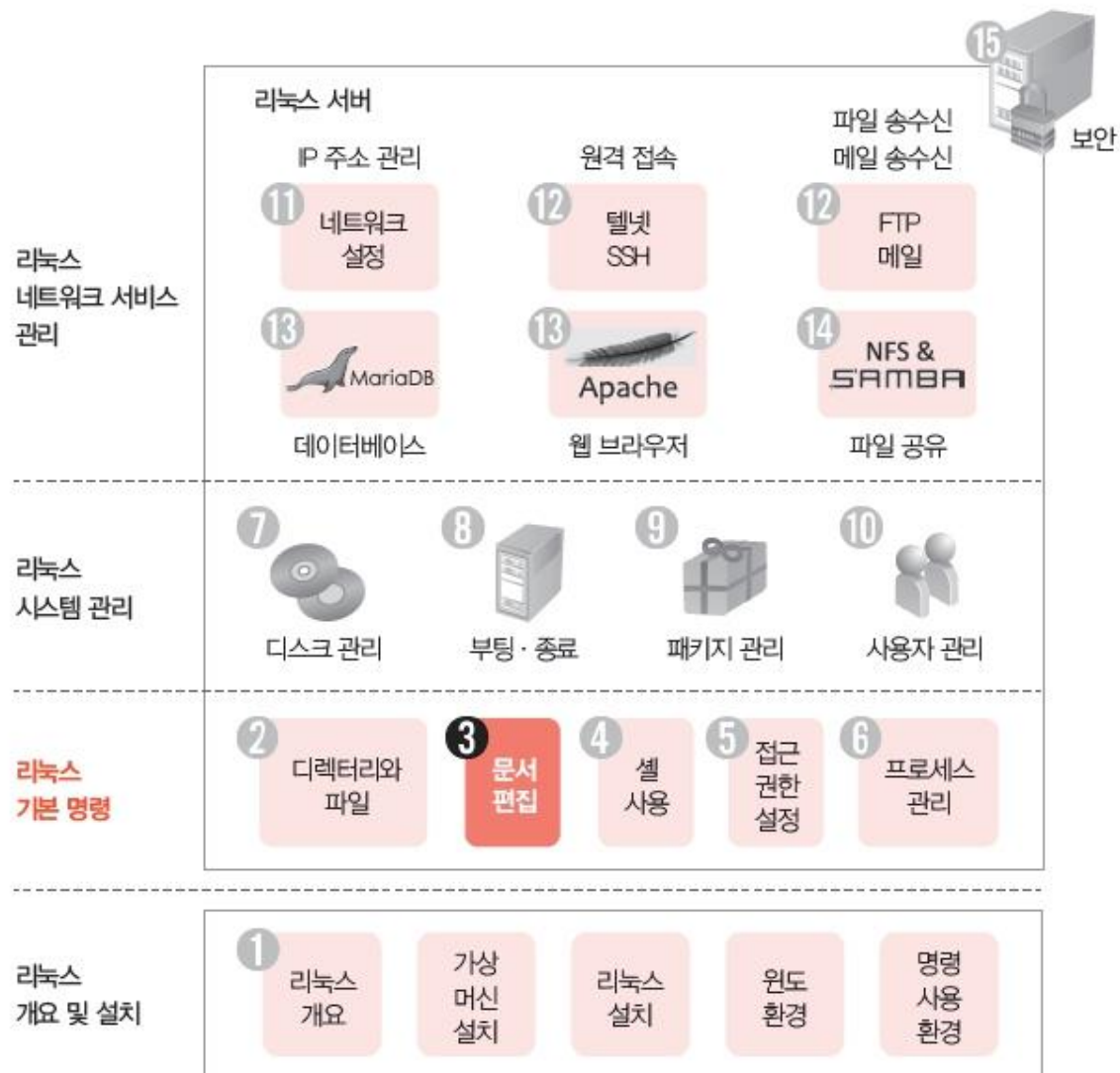
- 리눅스에서 사용하는 편집기의 종류를 알아본다.
- 대표적인 화면 편집기인 vi를 사용할 수 있다.
- vi의 환경을 설정할 수 있다.

00 개요

00. 개요

■ 리눅스 학습 맵에서 3장의 위치

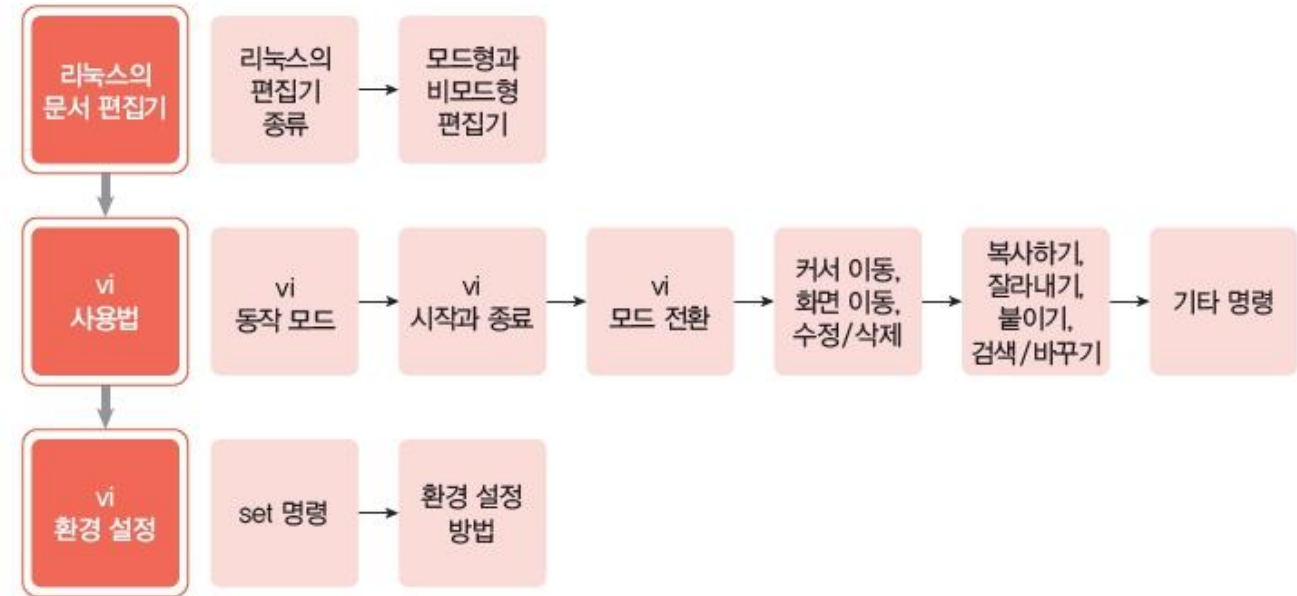
- 3장은 제2단계 중 두 번째 항목으로 리눅스의 대표적인 편집기인 vi를 이용하여 문서를 편집하고 파일을 만드는 방법을 익힌다.



00. 개요

■ 3장의 내용 구성

- 리눅스의 대표적인 문서 편집기인 vim을 학습
- vim은 유닉스에서 제공하는 편집기인 vi를 업그레이드 한 것으로 사용법은 동일
- vim(vi)를 이용하여 문서를 편집하는 방법 학습
- vim(vi)의 각 모드를 이해하고 명령 모드의 명령키 사용 방법과 마지막 행 모드의 사용 방법, vi 환경 설정 방법을 학습



01 리눅스의 문서 편집기

01. 리눅스의 문서 편집기

- GUI형 편집기
- 터미널에서 사용할 수 있는 편집기

표 3-1 리눅스의 편집기 종류

구분	종류
행 단위 편집기	ed, ex, sed
화면 단위 편집기	vi, emacs(이맥스)
GUI 편집기	gedit

01. 리눅스의 문서 편집기

■ 모드형과 비모드형 편집기

- 모드형: 입력 모드와 명령 모드를 구분
 - 입력 모드: 텍스트를 입력할 수 있는 모드
 - 명령 모드: 텍스트 수정/삭제/복사/붙이기 등 편집 모드
- 비모드형: 모드 구분이 없음: 한글, 워드 등
 - alt, ctrl, shift 등 특수 키를 이용하여 편집 명령 수행

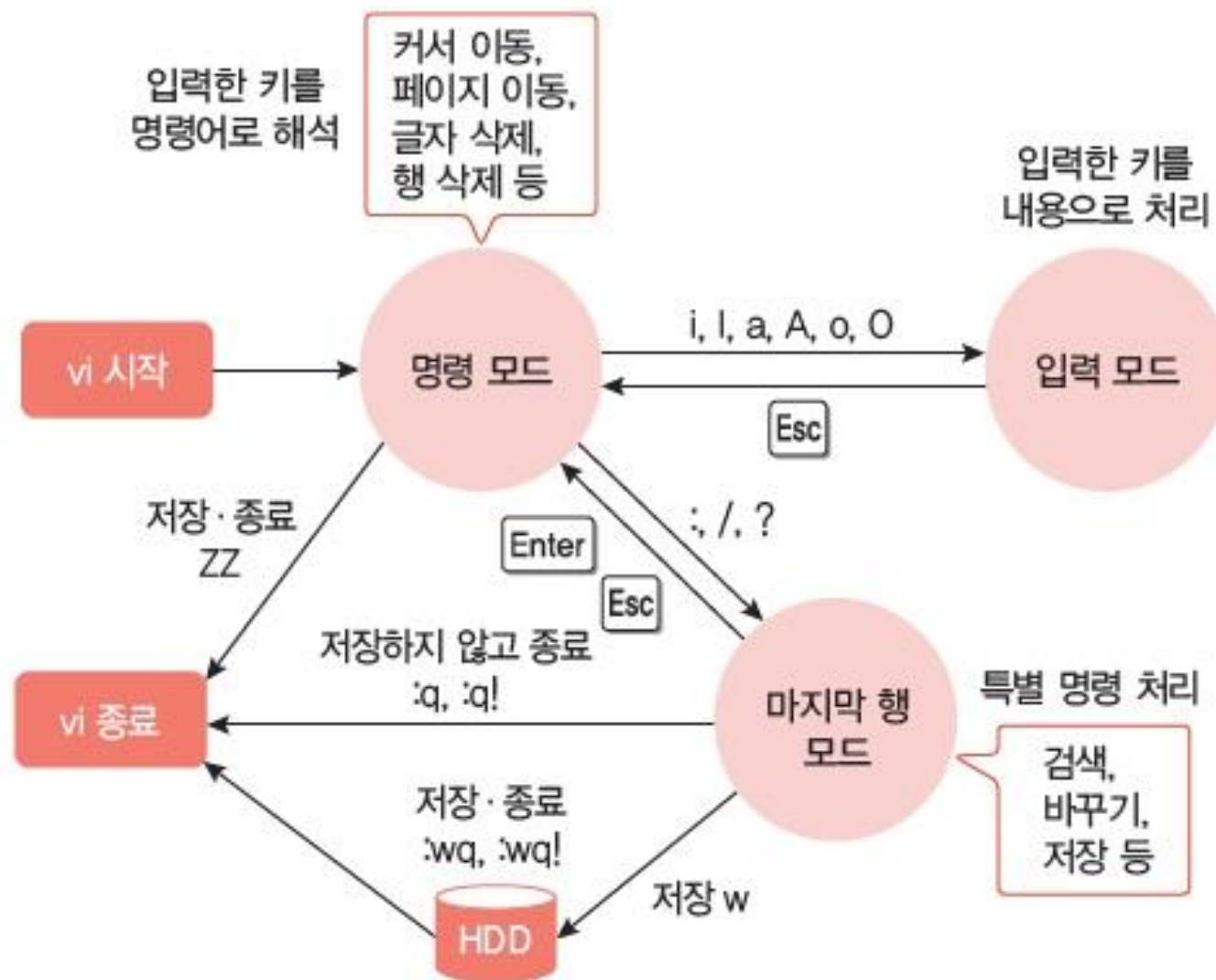
표 3-2 모드형과 비모드형 편집기의 비교

구분		모드형(vi)	비모드형(메모장)
입력 모드		텍스트 입력	
명령 모드의 예	복사	yy	Ctrl +C
	붙이기	p	Ctrl +V
	저장	:wq, ZZ	Ctrl +S
모드 전환		i, a, o, Esc	해당 없음

02 vi 사용법

02. vi 사용법

■ vi 동작 모드



02. vi 사용법

■ vi 시작과 종료

```
[user1@localhost ~]$ vi test.txt
```

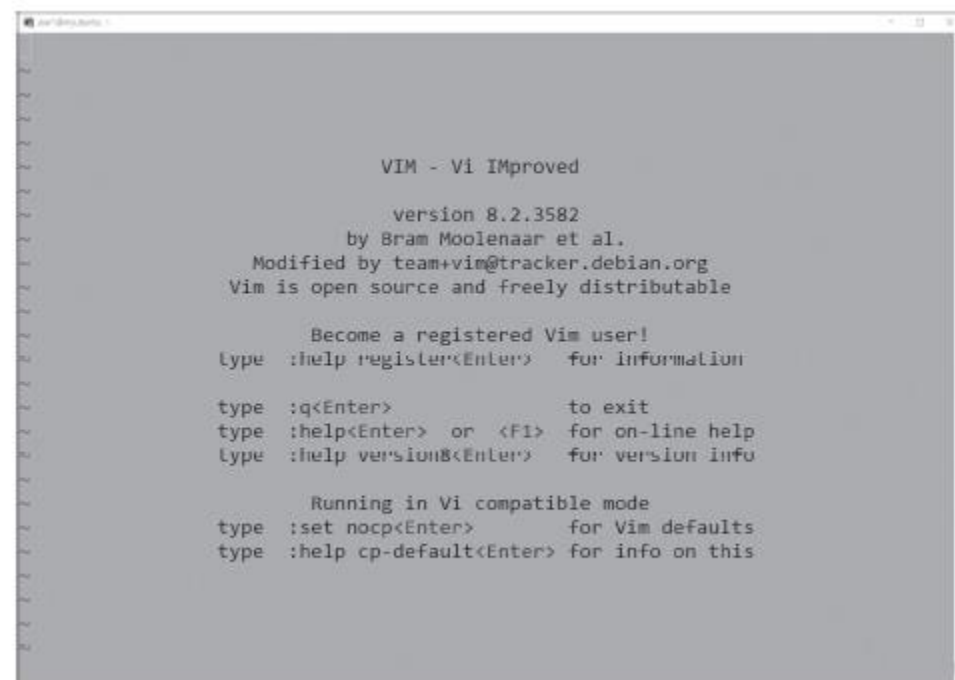
→ test.txt라는 파일이 열린다. 이 파일이 없으면 빈 파일이 열린다.

```
[user1@localhost ~]$ vi
```

→ 빈 파일이 열린다. 나중에 파일명을 정할 수 있다.

표 3-3 vi의 저장과 종료 명령키

모드	명령키	기능
마지막 행 모드	:q	vi에서 작업한 것이 없을 때 그냥 종료한다.
	:q!	작업한 내용을 저장하지 않고 종료한다.
	:w 파일명	작업한 내용을 저장만 한다. 파일명을 지정하면 다른 파일로 저장한다.
	:wq, :wq!	작업한 내용을 저장하고 vi를 종료한다.
명령 모드	ZZ(Shift +zz)	작업한 내용을 저장하고 vi를 종료한다.



02. vi 사용법

■ vi 모드 전환

- 명령 모드 -> 입력 모드 전환
- 입력 모드 -> 명령 모드 전환
 - Esc 키

표 3-4 입력 모드 전환 명령키

명령키	기능
i	현재 커서 자리에 입력한다.
a	현재 커서 다음 자리에 입력한다.
o	커서가 위치한 행의 다음 행에 입력한다.
I(대문자 i)	커서가 위치한 행의 첫 칼럼으로 이동하여 입력한다.
A	커서가 위치한 행의 마지막 칼럼으로 이동하여 입력한다.
O	커서가 위치한 행의 이전 행에 입력한다.

02. vi 사용법

■ i 명령키를 사용해 입력 모드로 전환하기

```
user1@myubuntu:~/linux_ex/ch3$ vi test.txt
```

```
ubuntu linux study
```

```
I like linux
```

```
~
```

```
~
```

```
~
```

→  키를 누르면 다음 행으로 이동한다.

→  키를 누르면 명령 모드로 전환된다.

```
ubuntu linux study
```

```
I like linux
```

```
~
```

```
~
```

```
~
```

→ 명령 모드로 전환되고 커서가 x 위로 이동한다.

02. vi 사용법

■ i와 a 명령키의 차이

```
ubuntu linux study
I like linu ubuntux ~
~
```

커서가 마지막 글자인 x에 있을 때
i키를 누르고 'ubuntu'를 입력할 경우

→ 명령 모드로 전환되고 커서가 u 위로 이동한다.

```
ubuntu linux study
I like linu ubuntu linux
~
```

커서가 u에 있을 때
a키를 누르고 'linu'를 입력할 경우

→ 명령 모드로 전환되고 커서가 u 위로 이동한다.

02. vi 사용법

■ o 명령키를 사용하여 입력 모드로 전환하기

```
ubuntu linux study
```

```
I like linu ubuntu linux
```

→ 명령 모드에서 o를 입력하면 커서가 아래 행으로 이동한다.

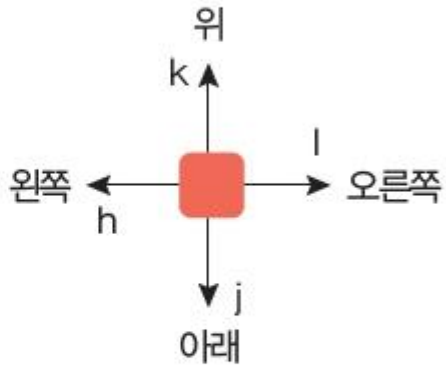

```
■
```

```
~
```


02. vi 사용법

■ 커서 이동하기

표 3-5 커서 이동 명령키

명령키	기능	
k / j	커서를 한 행 위 / 아래로 이동한다.	
l / h	커서를 한 글자 오른쪽 / 왼쪽으로 이동한다.	
^ 또는 0 / \$	커서를 현재 행의 처음 / 마지막으로 이동한다.	
- / + 또는 	커서를 앞 / 다음 행의 처음으로 이동한다.	
H / M / L	커서를 화면의 맨 위 / 중간 / 맨 아래 행으로 이동한다.	
w / e	커서를 다음 단어의 처음 / 마지막 글자로 이동한다.	
b	커서를 앞 단어의 첫 글자로 이동한다.	

02. vi 사용법

■ 커서 이동하기 예제

- 1 Come, have breakfast.
10
- 2 Look at my hands and my feet.
- 3 He saw and believed.
- 4 Who do you say that I am?
- 5 I do know him and I keep his word.
11
- 6 You always have the poor with word.
8 1
- 7 Forgive and you will be forgiven.
6 4 5 3 7
- 8 He loved his own to the end.
9 2
- 9 The Lord is with you.
12

커서가 7행 'will'의 첫 글자인 w에 있다고 가정

- ① k 명령키를 입력하면 커서가 w 바로 위에 있는 t로 이동한다.
- ② j 명령키를 입력하면 커서가 w 바로 아래에 있는 공백으로 이동한다.
- ③ w 명령키를 입력하면 커서가 다음 단어 be의 첫 글자인 b로 이동한다.
- ④ b 명령키를 입력하면 커서가 앞 단어 you의 첫 글자인 y로 이동한다.
- ⑤ e 명령키를 입력하면 커서가 현재 단어 will의 마지막 글자인 l로 이동한다. 여기서 다시 e 명령키를 입력하면 커서가 다음 단어 be의 마지막 글자인 e로 이동한다.
- ⑥ ^ (캐럿) 명령키(Shift+6)를 입력하면 커서가 현재 행의 첫 글자인 F로 이동한다.
- ⑦ \$ 명령키(Shift+4)를 입력하면 커서가 현재 행의 끝인 . (마침표)로 이동한다.
- ⑧ - 명령키를 사용하면 커서가 위행의 첫 글자인 Y로 이동한다.
- ⑨ + 명령키를 사용하면 커서가 아래 행의 첫 글자인 H로 이동한다.
- ⑩ H 명령키를 입력하면 커서가 화면의 첫 행으로 이동한다.
- ⑪ M 명령키를 입력하면 커서가 화면의 중간 행으로 이동한다.
- ⑫ L 명령키를 입력하면 커서가 화면의 마지막 행으로 이동한다. 내용이 화면 크기보다 작으면 내용의 마지막 행으로 커서가 이동한다.
- ⑬ h나 l (소문자 L) 명령키를 입력하면 커서가 좌우로 한 칸씩 움직인다.

02. vi 사용법

■ 화면 이동하기

표 3-7 화면 이동 명령키

기존 명령키	기능	추가 명령키
\wedge u(Ctrl +u)	반 화면 위로 이동한다.	
\wedge d(Ctrl +d)	반 화면 아래로 이동한다.	
\wedge b(Ctrl +b)	한 화면 위로 이동한다.	Page Up
\wedge f(Ctrl +f)	한 화면 아래로 이동한다.	Page Down
\wedge y(Ctrl +y)	화면을 한 행만 위로 이동한다.	
\wedge e(Ctrl +e)	화면을 한 행만 아래로 이동한다.	

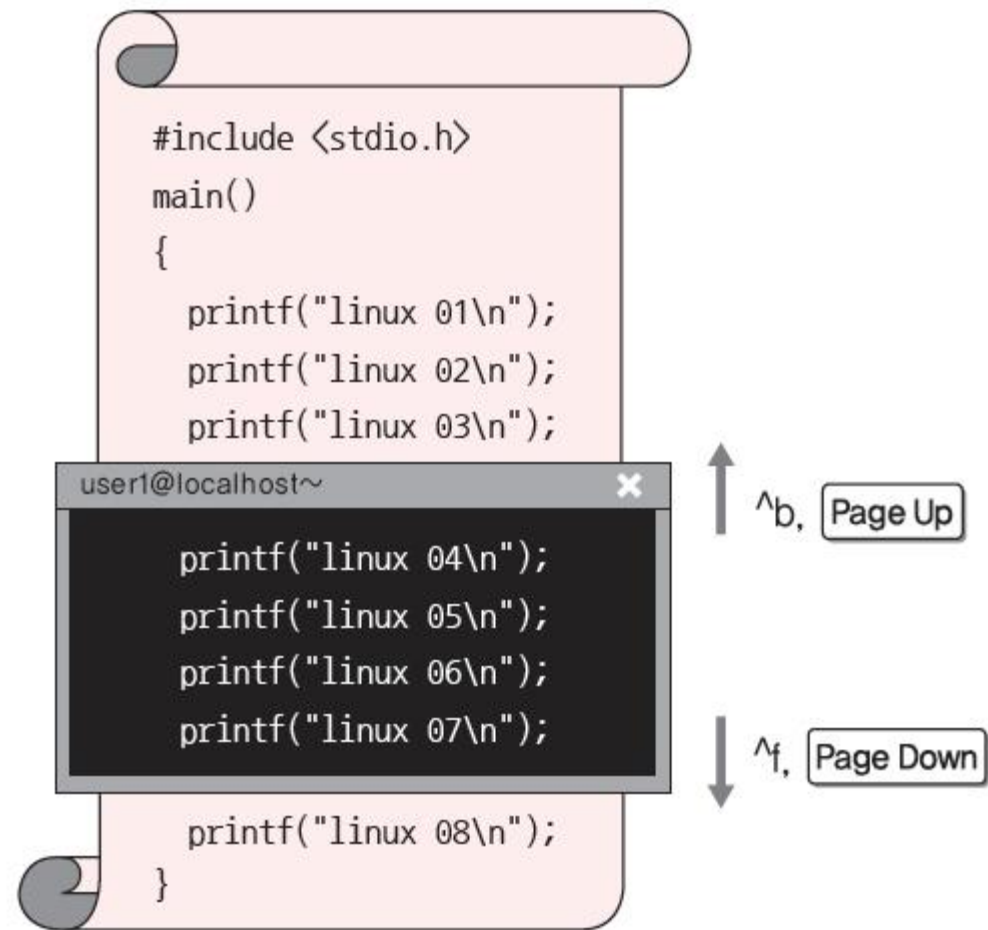


그림 3-4 화면 이동 명령키의 사용

02. vi 사용법

■ 특정 행으로 바로 이동하기


표 3-8 특정 행으로 바로 이동하는 명령키

명령키	기능
G(Shift +g)	파일의 마지막 행으로 커서가 이동한다.
행 번호G(Shift +g)	지정한 행 번호로 커서가 이동한다.
:행 번호	지정한 행 번호로 커서가 이동한다(마지막 행 모드).
:\$	파일의 마지막 행으로 커서가 이동한다(마지막 행 모드).

02. vi 사용법

■ 내용 수정하기

표 3-9 내용 수정 명령키

명령키	기능
r	커서가 위치한 글자를 다른 글자로 수정한다.
cw, #cw	커서 위치부터 현재 단어의 끝까지 수정한다. #에는 수정할 단어의 수를 지정한다. 예를 들어 3cw는 커서 위치부터 세 단어를 수정한다.
s, #s	커서 위치부터  를 입력할 때까지 수정한다. #에는 수정할 글자의 수를 지정한다. 예를 들어 5s는 커서 위치부터 다섯 글자를 수정한다.
cc	커서가 위치한 행의 내용을 모두 수정한다.
C	커서 위치부터 행의 끝까지 수정한다.

02. vi 사용법

■ 한 글자 수정하기: r 명령키

```
ubuntu winux study      → r 명령키로 글자를 수정한다(l→w)  
I like linu ubuntu linux  
  
~
```

■ 단어 수정하기: cw, #s 명령키

winux의 w에 커서를 두고 cw 또는 5s를 입력하여 수정 상태로 바꾸고 editor 입력후 Esc키 입력

```
ubuntu study      → cw 명령키 입력 시 winux가 사라진다.  
I like linu ubuntu linux
```

```
ubuntu editor study      → 수정 완료 후 Esc를 눌러야 명령 모드로 전환된다.  
I like linu ubuntu linux
```

02. vi 사용법

■ 행 단위 수정하기: C, cc 명령키

커서 위치부터 행의 끝까지 수정: C 명령키

```
ubuntu editor stud$  
I like linu ubuntu linux
```

→ 행의 맨 끝에 \$가 표시되고 입력 모드로 전환된다.

현재 행의 모든 내용 삭제하고 다시 작성: cc 명령키

```
■  
I like linu ubuntu linux
```

→ 모두 지워지고 행의 처음으로 이동한다. 입력 모드로 전환된다.

02. vi 사용법

■ 내용 삭제

표 3-10 내용 삭제 명령키

명령키	기능
x, #x	커서 위치의 글자를 삭제한다. #에는 삭제할 글자 수를 지정한다.
dw, #dw	커서 위치의 단어를 삭제한다. #에는 삭제할 단어 수를 지정한다.
dd, #dd	커서 위치의 행을 삭제한다. #에는 삭제할 행의 수를 지정한다.
D([Shift]+d)	커서 위치부터 행의 끝까지 삭제한다.

02. vi 사용법

글자 삭제

```
ubuntu editor vi study  
I like linu ubuntu linux
```

x 명령키로
l 삭제

```
ubuntu editor vi study  
I like inu ubuntu linux
```

여러 글자 삭제

```
ubuntu editor vi study  
I like ubuntu linux
```

dw로
inu 삭제

```
ubuntu editor vi study  
I like ubuntu linux
```

3x로
inu 삭제

행 삭제

```
ubuntu editor vi study  
I like           
```

커서 위치부터
행의 끝까지 삭제: D

```
ubuntu editor vi study  
          
```

행 전체 삭제: dd

02. vi 사용법

■ 명령 취소


표 3-11 이전 명령 취소 명령키

명령키	기능
u	명령을 취소한다.
U	해당 행에서 한 모든 명령을 취소한다.
:e!	마지막으로 저장한 내용 이후의 것을 버리고 새로 작업한다.

149p. 따라해보기 : vi로 입력,수정,삭제,복구하기

- ① vi로 새로운 파일인 exec.txt 파일을 연다
- ② i를 입력하여 입력 모드로 전환하고 내용을 입력
- ③ Esc키 를 입력하여 명령 모드로 전환하고 :w 명령으로 파일 내용을 저장
- ④ 커서를 3행의 beginner로 이동: 3G,I 또는 w
- ⑤ beginner를 expert로 수정: cw 또는 8s
- ⑥ - 명령키로 2행의 첫 글자인 N으로 이동
- ⑦ w 명령키로 커서를 meet로 이동하고, 단어 meet를 dw 명령키로 삭제
- ⑧ 단어 y부터 행의 끝까지 D 명령키로 삭제
- ⑨ U 명령키로 2행에서 한 모든 삭제를 취소
- ⑩ :wq 명령으로 파일 내용을 저장하고 종료

```
Good morning everyone.  
Nice to meet you.  
I am a linux beginner.  
Now introduce yourself.
```



```
Good morning everyone.  
Nice to meet you.  
I am a linux expert.  
Now introduce yourself.
```

02. vi 사용법

■ 복사 및 잘라 붙이기

표 3-12 복사하기, 잘라내기, 붙이기 명령키

명령키	기능
yy, #yy	커서가 위치한 행을 복사한다. #에는 복사할 행의 수를 지정한다.
p	커서가 위치한 행의 아래쪽에 붙인다.
P	커서가 위치한 행의 위쪽에 붙인다.
dd, #dd	커서가 위치한 행을 잘라둔다. 삭제와 같은 기능이다. #에는 잘라줄 행의 수를 지정한다.

02. vi 사용법

```
ubuntu editor vi study  
I like
```

```
ubuntu editor vi study  
I like
```

```
ubuntu editor vi study  
I like
```

```
ubuntu editor vi study  
ubuntu editor vi study  
I like  
I like
```

test.txt 파일 열기

- ① 2yy 명령키로 두 행을 모두 복사
- ② 아래 행으로 이동하여 p 명령키로 붙이기

- ① dd로 3행 잘라내기
- ② 커서를 2행으로 이동하여 P 명령키로 붙이기

02. vi 사용법

■ 네임드 버퍼 사용하기

- 언네임드 버퍼: 이름을 붙이지 않은 버퍼
- 네임드 버퍼: 이름을 붙여서 사용할 수 있는 버퍼(문자 또는 숫자)
 - 네임드 버퍼: "a, "b, "c, "d, ..., "z
 - 숫자 버퍼: "1, "2, ..., "9

```
ubuntu editor vi study
ubuntu editor vi study
I like
I like
```

2행을 잘라서 "a 버퍼에 저장
"add

```
ubuntu editor vi study
I like
I like
```

2행을 잘라서 "b 버퍼에 저장
"bdd

```
I like
ubuntu editor vi study
ubuntu editor vi study
I like
```

- ① 커서를 1행으로 이동
- ② "b 버퍼의 내용을 1행 위 행에 붙여넣기: "bP
- ③ "a 버퍼의 내용을 1행 아래 행에 붙여넣기: "aP

02. vi 사용법

■ 마지막 행 모드에서 복사하기, 잘라내기, 붙이기

- :을 입력하여 마지막 행 모드로 전환
- 범위 지정하기

표 3-13 범위 지정 명령키

명령키	기능
1, \$ 또는 %	1행부터 마지막 행까지 지정한다.
1, .	1행부터 커서가 있는 행까지 지정한다.
., \$	커서가 있는 행부터 마지막 행까지 지정한다.
., -3	현재 행과 이전 세 행까지(총 네 행) 지정한다.
10, 20	10행부터 20행까지 지정한다.

TIP 명령키에서 여러 행을 지정하는 경우, 숫자와 숫자 사이의 ,(쉼표)는 띄어쓰기를 하지 않는다.

02. vi 사용법

- 복사하기, 잘라내기, 붙이기

표 3-14 마지막 행 모드에서의 복사하기, 잘라내기, 붙이기 명령키

명령키	기능
:#y	#로 지정한 행을 복사한다. 예를 들면 3y는 세 번째 행을 복사한다.
:<범위>y	범위로 지정한 행을 복사한다. 예를 들면 2,4y는 2~4행을 복사한다.
:#d	#로 지정한 행을 잘라낸다(삭제). 예를 들면 3d는 세 번째 행을 잘라낸다.
:<범위>d	범위로 지정한 행을 잘라낸다(삭제). 예를 들면 1,4d는 1~4행을 잘라낸다.
:pu	현재 행 다음에 버퍼의 내용을 붙인다.
:#pu	#로 지정한 행 다음에 버퍼의 내용을 붙인다. 예를 들면 4pu와 같이 지정한다.

02. vi 사용법

```
I like
uubuntu editor vi study
ubuntu editor vi study
I like
~
```

현재 커서는 2행에 위치

```
I like
I like
ubuntu editor vi study
uubuntu editor vi study
~
```

2행,3행 잘라내기 :2,3d
2행 다음에 붙이기 :2pu

```
I like
I like
ubuntu editor vi study
ubuntu editor vi study
I like
I like
~
```

1,2행 복사하기 :1,2y
4행 다음에 붙이기 :4pu

02. vi 사용법

■ 검색하기

표 3-15 검색 명령키

명령키	기능
/문자열	문자열을 아래 방향으로 검색한다.
?문자열	문자열을 위 방향으로 검색한다.
n	원래 찾던 방향으로 다음 문자열을 검색한다.
N	역방향으로 다음 문자열을 검색한다.

02. vi 사용법

```
I like
I like
ubuntu editor vi study
ubuntu editor vi study
I like
I like
~
(생략)
/search hit BOTTOM, continuing at TOP
```

n 입력

```
I like
I like
ubuntu editor vi study
ubuntu editor vi study
I like
I like
~
(생략)
/ubuntu
```

- ①커서는 6행에 위치
- ②검색하려고 /을 입력하면 커서가 마지막 행으로 이동
- ③검색할 문자열인 'ubuntu'를 입력
- ④Enter를 누르면 6행이 파일의 마지막 행이므로 'search hit BOTTOM, continuing at TOP'이라는 메시지를 출력
- ⑤3행의 ubuntu로 커서 이동

계속 ubuntu를 검색하려고 n을 입력하면 4행의 ubuntu로 커서가 이동

02. vi 사용법

■ 바꾸기

표 3-16 바꾸기 명령키

명령키	기능
:s/문자열1/문자열2/	커서가 위치한 행에서 첫 번째로 나오는 문자열1을 문자열2로 바꾼다.
:%s/문자열1/문자열2/g	파일 전체에서 모든 문자열1을 문자열2로 바꾼다.
:<범위>s/문자열1/문자열2/	범위 내 모든 각 행에서 첫 번째로 나오는 문자열1을 찾아 문자열2로 바꾼다.
:<범위>s/문자열1/문자열2/g	범위 내 모든 행에서 문자열1을 문자열2로 바꾼다.
:<범위>s/문자열1/문자열2/gc	범위 내 모든 행에서 문자열1을 문자열2로 바꿀 때 수정할지 여부를 묻는다.

02. vi 사용법

```
I LIKE
I like
ubuntu editor vi study
ubuntu editor vi study
I like
I like
~
(생략)
:s/like/LIKE/
```

1행의 like를 LIKE로 바꾸기

```
I LIKE
I like
ubuntu ubuntu vi study
ubuntu ubuntu vi study
I like
I like
~
(생략)
:3,4s/editor/ubuntu/
```

3,4행의 editor를
ubuntu로 바꾸기

```
I LIKE
I like
UBUNTU ubuntu vi study
UBUNTU UBUNTU vi study
I like
I like
~
(생략)
:s/ubuntu/UBUNTU/g
```

3행- :s/ubuntu/UBUNTU/
4행- :s/ubuntu/UBUNTU/g

```
I LIKE
I LIKE
UBUNTU ubuntu vi study
UBUNTU UBUNTU vi study
I LIKE
I LIKE
~
(생략)
3 substitutions on 3 lines
```

like 전체를LIKE로 바꾸기
:%s/like/LIKE/g

160p. 따라해보기 : vi 편집 방법 익히기

- ① vi로 새로운 파일인 exec2.txt 파일 열기
- ② i를 입력하여 입력 모드로 전환하고 내용을 입력
- ③ Esc키 를 입력하여 명령 모드로 전환하고 :w 명령으로 파일 내용을 저장
- ④ 1행을 복사하여 3행 다음에 붙이기
- ⑤ 파일에서 morning이란 단어를 검색
- ⑥ 파일에서 morning이란 단어를 모두 찾아 afternoon으로 바꾸기
- ⑦ :wq 명령으로 파일 내용을 저장하고 종료

```
Good morning everyone.  
My name is Gildong Hong.  
This is a living room.
```



```
Good afternoon everyone.  
My name is Gildong Hong.  
This is a living room.  
Good afternoon everyone.
```

02. vi 사용법

■ 파일 읽어오기, 여러 파일 편집하기

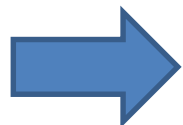
표 3-17 파일 관련 명령키

명령키	기능
:r 파일	지정한 파일을 읽어들이어 현재 커서 위치에 삽입한다.
:e 파일	지정한 파일로 전환한다(기존 파일을 :w로 저장한 뒤에 실행해야 한다).
:n	vi 시작 시 여러 파일을 지정했을 경우 다음 파일로 작업을 이동한다.

02. vi 사용법

■ 다른 파일 읽어오기

```
I LIKE  
I LIKE  
UBUNTU ubuntu vi study  
UBUNTU UBUNTU vi study  
I LIKE  
I LIKE
```



```
I LIKE  
I LIKE  
UBUNTU ubuntu vi study  
UBUNTU UBUNTU vi study  
Good afternoon everyone.  
This is a living room.  
My name is Gildong Park.  
This is a living room.  
My name is Gildong Hong.  
Good afternoon everyone.  
I LIKE  
I LIKE
```

:r exec2.txt로 파일
읽어오기

- ① test.txt 파일 열기
- ② 4행으로 이동

02. vi 사용법

■ 파일 편집을 마치고 다른 파일 편집하기

```
I LIKE  
I LIKE  
UBUNTU ubuntu vi study  
UBUNTU UBUNTU vi study  
Good afternoon everyone.  
This is a living room.  
(생략)  
~
```

```
E37: No write since last change (add ! to override)
```

:e 명령키로 다른 파일 편집 - :e exec.txt

저장하지 않고 :e exec.txt를 하면 오류발생
:w 저장하고 또는 :e! exec.txt로 수행해야함

02. vi 사용법

■ 여러 파일 편집하기

```
user1@myubuntu:~/linux_ex/ch3$ vi test.txt exec.txt exec2.txt
```

동시에 여러 개 파일명 지정 가능

다음 파일 작업을 하려면 :n으로 이동

02. vi 사용법

■ vi에서 셸 명령 사용하기

표 3-18 셸 명령 실행 명령키

명령키	기능
:! 셸 명령	vi 작업을 잠시 중단하고 셸 명령을 실행한다(vi로 돌아오려면 Enter, ↵ 를 누른다).
:sh	vi를 잠시 빠져나가서 셸 명령을 실행한다(vi로 돌아오려면 exit 명령을 입력한다).

02. vi 사용법

- :! 명령키 이용하기

(생략)

~

~

:! ls



```
user1@myubuntu:~/linux_ex/ch3$ vi test.txt
```

```
exec.txt    exec2.txt    test.txt
```

```
Press ENTER or type command to continue
```

02. vi 사용법

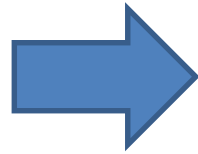
- :sh 명령키 이용하기

(생략)

~

~

:sh



```
user1@myubuntu:~/linux_ex/ch3$ ls
exec.txt      exec2.txt     test.txt
user1@myubuntu:~/linux_ex/ch3$ exit
```

vi로 돌아가려면 exit를 입력

02. vi 사용법

■ 기타 명령키

표 3-19 기타 명령키

명령키	기능
Ctrl +l(소문자 L)	현재 화면을 다시 출력한다.
Ctrl +g	현재 커서 위치의 행 번호를 마지막 행에 출력한다.
Shift +j(대문자 J)	현재 행과 아래 행을 연결하여 한 행으로 만든다.
.(마침표)	바로 직전에 했던 명령을 반복한다.
~(틸드)	커서 위치의 글자를 대문자나 소문자로 바꾼다.

02. vi 사용법

- 화면 다시 출력하기: `ctrl+l`
- 행 연결하기: `J`

```
I like
I like
UBUNTU ubuntu vi study
(생략)
~
```

J 입력

```
I LIKE I LIKE
UBUNTU ubuntu vi study
(생략)
~
```

- 이전 명령 반복하기: `.(점)`

```
I LIKE I LIKE
UBUNTU ubuntu vi study
(생략)
~
```

. 입력

```
I LIKE I LIKE UBUNTU ubuntu vi study
(생략)
~
```

- 대소문자 바꾸기: `~`

```
I LIKE I LIKE UBUNTU
Ubuntu vi study
(생략)
~
```

커서를 u로 옮기고 ~ 명령키를
입력하면 U로 바뀜

168p. 따라해보기 : 기타 유용한 명령 익히기

- ① vi로 새로운 파일인 exec3.txt 파일 열기
- ② i를 입력하여 입력 모드로 전환하고 내용을 입력
- ③ Esc키 를 입력하여 명령 모드로 전환하고 :w 명령으로 파일 내용을 저장
- ④ 3행 다음에 exec2.txt 파일을 읽어들이
- ⑤ 1행과 2행을 연결하여 한 행으로 만듦
- ⑥ 반복 명령키로 1행과 2행을 연결하여 한 행으로 만듦
- ⑦ ~명령키로 1행의 everyone을 EVERYONE으로 바꿈
- ⑧ exec3.txt 파일을 저장하고, 셸 명령을 실행하여 현재 디렉터리의 경로를 확인
- ⑨ :wq 명령으로 파일 내용을 저장하고 종료

```
Good  
morning  
everyone.  
~
```

```
Good morning EVERYONE.  
Good afternoon everyone.
```


03 vi 환경 설정

03 vi 환경 설정

표 3-20 vi 환경 설정 명령

set 명령과 옵션	기능
set nu	파일 내용의 각 행에 행 번호를 표시한다(보이기만 할 뿐 저장되지는 않는다).
set nonu	행 번호를 감춘다.
set list	눈에 보이지 않는 특수문자를 표시한다(tab:↵, eol:\$ 등).
set nolist	특수문자를 감춘다.
set showmode	현재 모드를 표시한다.
set noshowmode	현재 모드를 감춘다.
set	set으로 설정한 모든 vi 환경 설정 값을 출력한다.
set all	모든 vi 환경 변수와 현재 값을 출력한다.

03 vi 환경 설정

■ 행 번호 표시 - :set nu

파일: exec2.txt	set nu 실행 후
Good afternoon everyone. This is a living room. My name is Gildong Park. This is a living room. My name is Gildong Hong. Good afternoon everyone. ~ :set nu	1 Good afternoon everyone. 2 This is a living room. 3 My name is Gildong Park. 4 This is a living room. 5 My name is Gildong Hong. 6 Good afternoon everyone. ~ :set nu

03 vi 환경 설정

■ 특수문자 표시 - :set list

파일: exec2.txt	set list 실행 후
Good afternoon everyone. This is a living room. My name is Gildong Park. This is a living room. My name is Gildong Hong. Good afternoon everyone. ~ :set list	Good afternoon everyone.\$ This is a living room.\$ My name is ^I Gildong Park.\$ This is a living room.\$ My name is Gildong Hong.\$ Good afternoon everyone.\$ ~ :set list

03 vi 환경 설정

■ 환경 설정 값 표시- :set

```
~  
:set  
— Options —  
scroll=11          ttyfast          ttymouse=xterm  
fileencoding=utf-8  
fileencodings=ucs-bom,utf-8,default,latin1  
runtimepath=~/.vim,/var/lib/vim/addons,/usr/share/vim/vimfiles,/usr/share/vim/  
vim82,/usr/share/vim/vimfiles/after,/var/lib/vim/addons/after,~/.vim/after  
Press ENTER or type command to continue
```

03 vi 환경 설정

■ 모든 환경 변수 표시- :set all

```
:set all
(생략)
isfname=@,48-57,/.,-,_+,.,#,$,%~, =
isident=@,48-57,_,192-255
iskeyword=@,48-57,_,
matchpairs=(:),{:,[:]
maxmempattern=1000
nrformats=bin,octal,hex
packpath=~/.vim,/var/lib/vim/addons,/etc/vim,/usr/share/vim/vimfiles,/usr/share/vim/vim82,/usr/share/vim/vimfiles/after,/etc/vim/after,/var/lib/vim/addons/after,~/.vim/after
paragraphs=IPLPPPQPP TPHPLIPpLpItpplpipbp
path=.,/usr/include,,
runtimepath=~/.vim,/var/lib/vim/addons,/usr/share/vim/vimfiles,/usr/share/vim/vim82,/usr/share/vim/vimfiles/after,/var/lib/vim/addons/after,~/.vim/after
scrollopt=ver,jump
sections=SHNHH HUnhsh
selection=inclusive
shellredir=%s 2>&1
suffixes=.bak,~,.,o,.h,.info,.swp,.obj
Press ENTER or type command to continue
```

03 vi 환경 설정

■ .exrc 파일에 환경 설정

- 사용자 홈 디렉터리에 .exrc 파일로 vi 환경설정 저장
- 모든 파일에 동일하게 적용

```
set nu  
set list  
set showmode
```

■ EXINIT 환경 변수에 설정하기

- 셸의 환경 변수인 EXINIT에 vi 환경 설정 가능

```
user1@myubuntu:~/linux_ex/ch3$ EXINIT='set nu list'  
user1@myubuntu:~/linux_ex/ch3$ export EXINIT
```

Thank You !