

IT CookBook CentOS 리눅스

시스템 & 네트워크

[강의교안 이용 안내]

- 본 강의교안의 저작권은 한빛아카데미(주)에 있습니다.
- 이 자료를 무단으로 전제하거나 배포할 경우 저작권법 136조에 의거하여 최고 5년 이하의 징역 또는 5천만원 이하의 벌금에 처할 수 있고 이를 병과(併科)할 수도 있습니다.



CentOS 리눅스

시스템 & 네트워크

Chapter 03. 문서 편집

목차

00. 개요

01. 리눅스의 문서 편집기

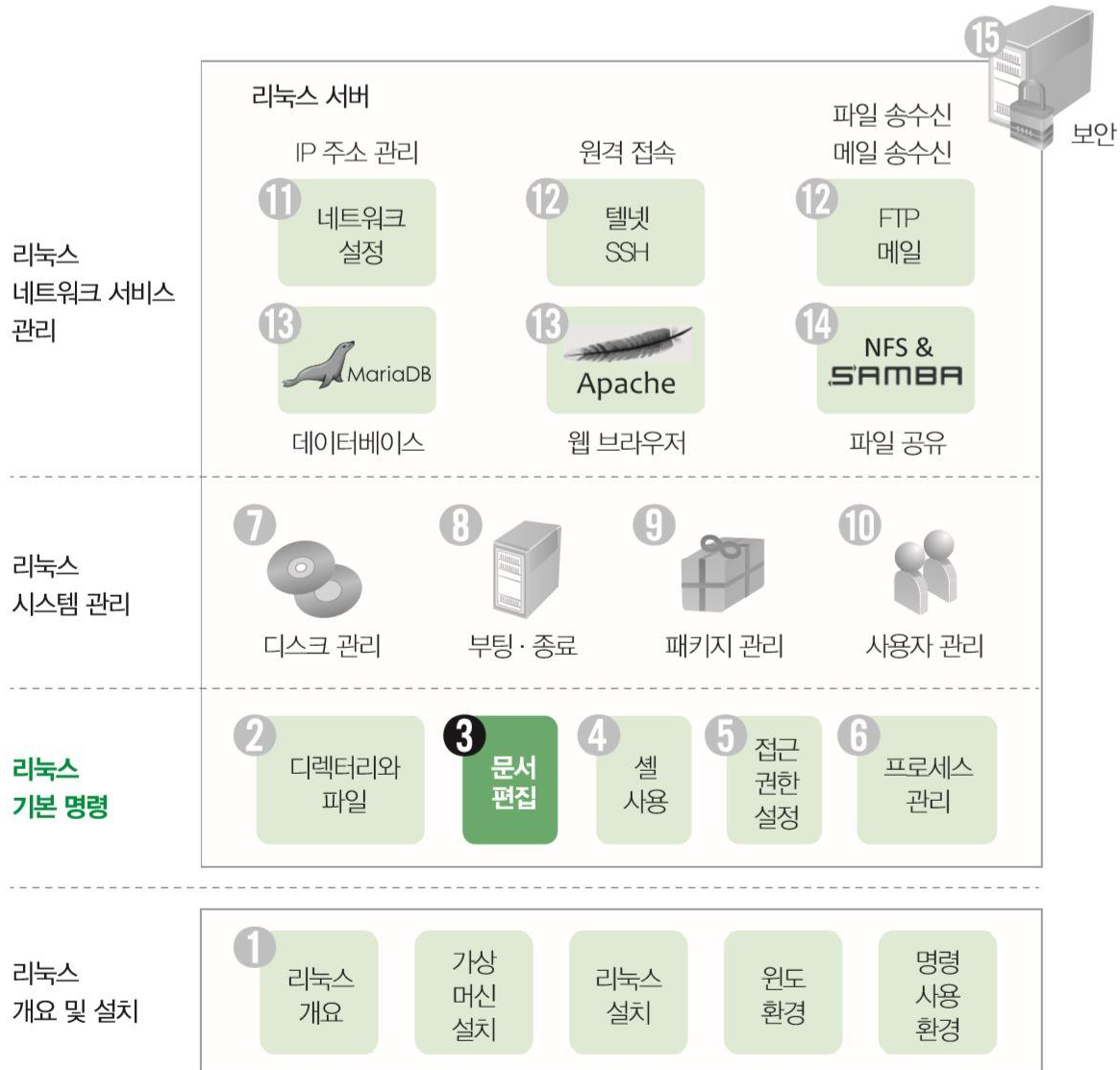
02. vi 사용법

03. vi 환경 설정

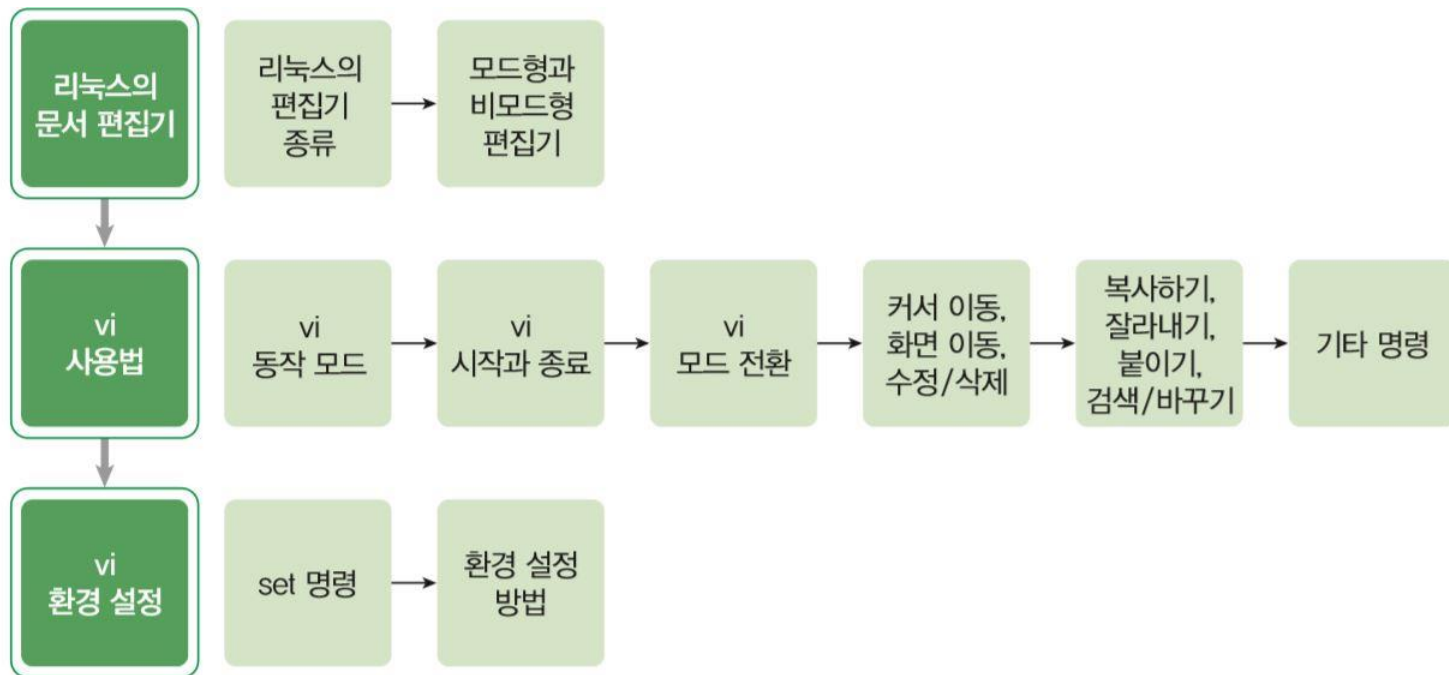
학습목표

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

00.개요



00.개요



01.리눅스의 문서 편집기

■ 리눅스의 편집기 종류

- GUI 환경인 그놈에서 제공하는 gedit
- 유닉스에서부터 사용했던 행 편집기(라인 편집기)와 화면 편집기

표 3-1 리눅스의 편집기 종류

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

■ 행 단위 편집기

- ed : 유닉스 초기의 행 편집기로 사용이 불편하여 거의 사용 않음
- ex : 행 편집기이지만 단독으로 사용하기보다는 vi에 연결하여 vi를 더욱 강력하게 하는 다양한 기능을 제공
- sed : 스트림 편집기로, 일반 편집기와 달리 지시된 명령에 따라 파일의 내용을 일괄적으로 바꿔서 출력해줌

■ 화면 단위 편집기

- vi : 리눅스에서 일반적으로 사용할 수 있는 화면 편집기
- emacs(이맥스) : 제공하는 기능이 매우 다양하지만 사용법이 어렵고 복잡하여 전문적인 애호가 위주로 사용
(GNU Emacs는 무료로 배포되며, 별도로 설치해야 함)

01.리눅스의 문서 편집기

■ 모드형과 비모드형 편집기

■ 모드형

- 입력 모드와 명령 모드가 구분
- 입력 모드 : 텍스트를 입력할 수 있는 모드
- 명령 모드 : 텍스트를 수정하고, 삭제하고, 복사와 붙이기 등 편집을 하는 모드
- 같은 글자라도 입력 모드에서는 텍스트로 처리하여 입력되고, 명령 모드에서는 텍스트로 입력되는 것이 아니라 편집 명령으로 사용
- vi는 모드형 편집기

■ 비모드형

- 입력 모드와 명령 모드가 구분되어 있지 않음
- 편집 기능을 Ctrl이나 Alt 같은 특수 키와 함께 사용
- 한글과 워드는 비모드형 편집기

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

■ vi 동작 모드

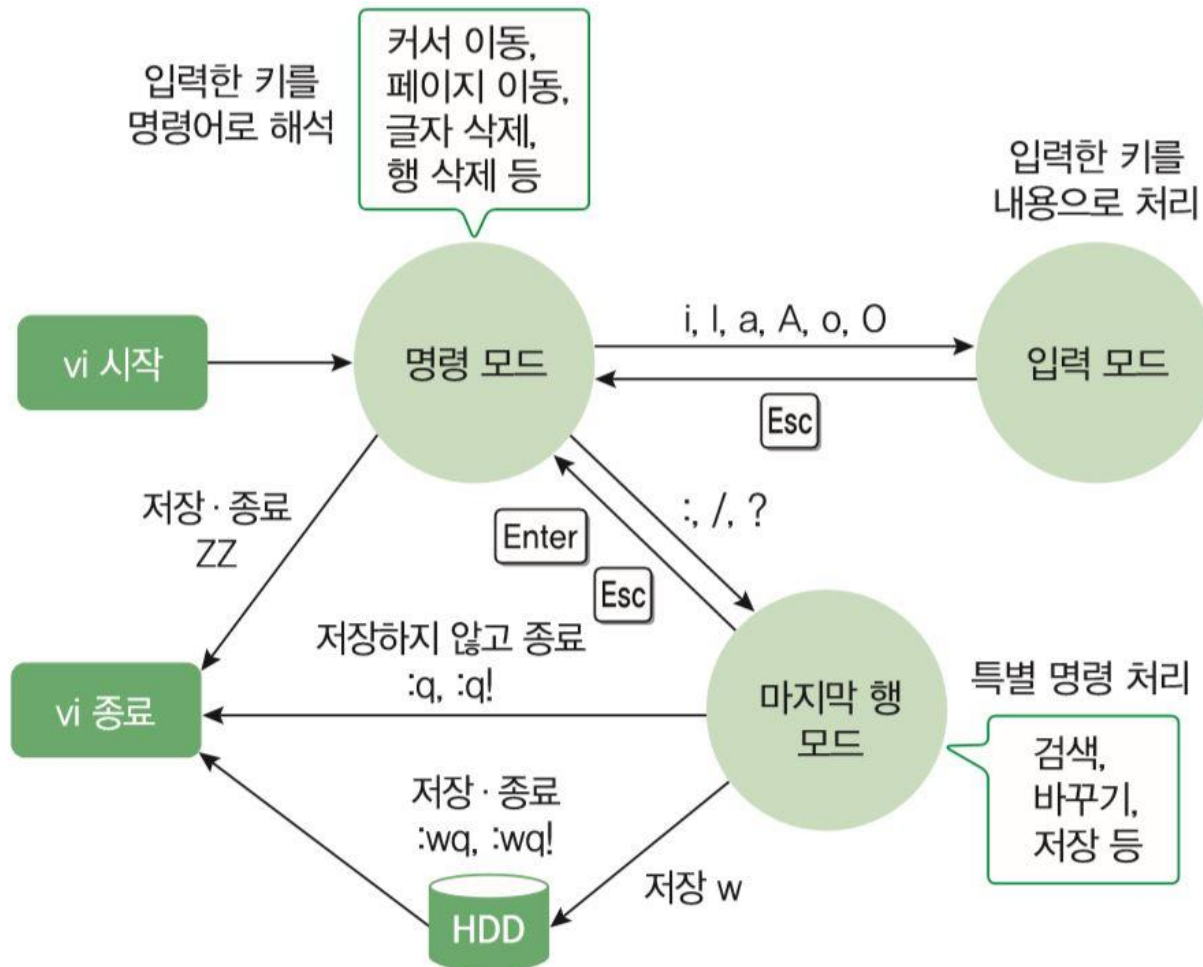


그림 3-1 vi 동작 모드

02.vi 사용법

■ vi 시작과 종료

vi

- **기능** 지정한 파일을 편집한다. 파일명을 지정하지 않으면 빈 파일이 열리고, 이 빈 파일의 파일명은 별도로 정할 수 있다.
- **형식** vi [파일]

■ vi 시작하기

- 파일을 지정할 경우: 해당 파일이 있으면 파일의 내용이 보이고, 없는 파일이면 빈 파일이 열린다.
- 파일을 지정하지 않을 경우: 그냥 빈 파일이 열린다(파일명은 저장할 때 지정 가능)

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

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

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

→ 빈 파일이 열린다. 나중에 파일명을 정할 수 있다.

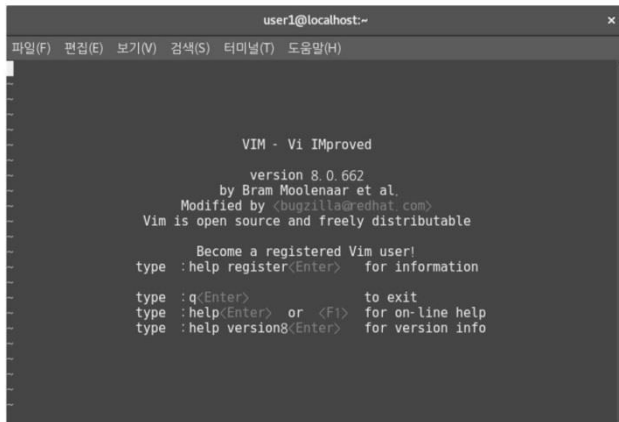


그림 3-2 vi의 초기 화면

02.vi 사용법

■ vi 시작과 종료

■ vi 파일 저장하고 종료하기

- 명령 모드나 마지막 행 모드에서 저장하고 종료 가능
- 파일의 저장과 종료를 동시에 수행할 수 있는 명령키 [shift+zz]
- 행 모드의 명령에서 !(느낌표)는 강제의 의미

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

- vi는 처음에 명령 모드로 시작하므로 내용을 입력하려면 입력 모드로 전환해야 함

표 3-4 입력 모드 전환 명령키

명령키	기능
i	커서 앞에 입력한다(현재 커서 자리에 입력한다).
a	커서 뒤에 입력한다(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

■ vi 모드 전환

- i 명령키를 사용해 입력 모드로 전환하기

- ch3 폴더를 만들고 그 아래의 test.txt 파일에서 작업

```
[user1@localhost ch3]$ vi test.txt
```

- vi를 실행한 뒤 명령 모드에서 i 명령 키를 입력하고 나서 다음 내용을 입력

```
CentOS linux study      → Enter 키를 누르면 다음 행으로 이동한다.  
I like linux           → Esc 키를 누르면 명령 모드로 전환된다.  
~  
~  
~
```

- 입력 모드에서 다시 명령 모드로 전환하기 위해 Esc 키를 누르면 커서가 x 위로 이동

```
CentOS linux study  
I like linuxx          → 명령 모드로 전환되고 커서가 x 위로 이동한다.  
~  
~  
~
```

02.vi 사용법

■ vi 모드 전환

■ i와 a 명령키의 차이

- 명령키 i는 커서 이전 위치에, a는 커서 다음 위치에 입력
- 커서가 마지막 글자인 x 위에 있는 상태에서 i를 입력하고 [Space bar]+CentOS를 입력하면 커서의 위치인 x의 이전에 CentOS가 입력

```
CentOS linux study
I like linu CentOSx      → 명령 모드로 전환되고 커서가 S 위로 이동한다.
~
~
~
```

- a 키로 입력 모드로 전환한 후 [Space bar]+linu를 입력하면 커서가 위치한 S 다음에 글자가 입력
- esc를 누르면 명령 모드로 전환되고 입력한 글자 중 마지막 글자인 u 위에 커서가 놓임

```
CentOS linux study
I like linu CentOS linu  → 명령 모드로 전환되고 커서가 u 위로 이동한다.
~
~
~
```

02.vi 사용법

■ vi 모드 전환

- o 명령키를 사용하여 입력 모드로 전환
 - 명령 모드에서 현재 커서가 위치한 후 다음 행에 글자를 입력하려면 o를 누름
 - esc를 누르면 다시 명령 모드로 전환되고 커서는 아랫 행의 첫 칼럼 자리에 위치

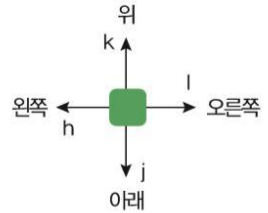

```
CentOS linux study
I like linu CentOS linux  → 명령 모드에서 o를 입력하면 커서가 아랫 행으로 이동한다.
~
~
```

02.vi 사용법

■ 커서와 화면 이동

■ 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 첫 / 마지막 글자로 이동한다.	
b	커서를 앞 단어의 첫 글자로 이동한다.	

02.vi 사용법

■ 커서와 화면 이동

■ 커서 이동하기

- ① k 명령키를 입력하면 커서가 w 바로 위에 있는 t로 이동
- ② j 명령키를 입력하면 커서가 w 바로 아래에 있는 공백으로 이동
- ③ w 명령키를 입력하면 커서가 다음 단어 be의 첫 글자인 b로 이동
- ④ b 명령키를 입력하면 커서가 앞 단어 you의 첫 글자인 y로 이동
- ⑤ e 명령키를 입력하면 커서가 현재 단어 will의 마지막 글자인 l로 이동
- ⑥ 만약 다시 e 명령키를 입력하면 커서가 다음 단어 be의 마지막 글자인 e로 이동
- ⑦ ^ (캐럿) 명령키([shift]+6)를 입력하면 커서가 현재 행의 첫 글자인 F로 이동
- ⑧ - 명령키를 사용하면 커서가 윗행의 첫 글자인 Y로 이동
- ⑨ + 명령키를 사용하면 커서가 아랫행의 첫 글자인 H로 이동
- ⑩ H 명령키를 입력하면 커서가 화면의 첫 행으로 이동 M 명령키를 입력하면 커서가 화면의 중간 행으로 이동
- ⑪ L 명령키를 입력하면 커서가 화면의 마지막 행으로 이동, 내용이 화면 크기보다 작으면 내용의 마지막 행으로 커서가 이동
- ⑫ h나 l(소문자 l) 명령키를 입력하면 커서가 좌우로 한 칸씩 이동

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

그림 3-3 커서 이동 명령키의 예

02.vi 사용법

■ 커서와 화면 이동

■ 커서 이동하기

- 초기의 유닉스 vi : 화살표 키로 커서를 이동할 수 없었음
- 리눅스 vi : 화살표 키로도 커서를 이동할 수 있음

표 3-6 기존 vi 명령키와 방향키, Home · End

명령키	방향키	명령키	방향키, Home · End
k	위 방향키(↑)	h	왼쪽 방향키(←)
j	아래 방향키(↓)	^ 또는 0	Home
l	오른쪽 방향키(→)	\$	End

02.vi 사용법

■ 화면 이동하기

- 파일 크기가 터미널의 화면 크기보다 클 경우 화면을 이동해야 함
- 아래 명령어 외에 [page up], [page down] 도 화면이동이 가능

표 3-7 화면 이동 명령키

기존 명령키	기능	추가 명령키
<code>^u</code> (<code>Ctrl</code> + <code>u</code>)	반 화면 위로 이동한다.	
<code>^d</code> (<code>Ctrl</code> + <code>d</code>)	반 화면 아래로 이동한다.	
<code>^b</code> (<code>Ctrl</code> + <code>b</code>)	한 화면 위로 이동한다.	<code>Page Up</code>
<code>^f</code> (<code>Ctrl</code> + <code>f</code>)	한 화면 아래로 이동한다.	<code>Page Down</code>
<code>^y</code> (<code>Ctrl</code> + <code>y</code>)	화면을 한 행만 위로 이동한다.	
<code>^e</code> (<code>Ctrl</code> + <code>e</code>)	화면을 한 행만 아래로 이동한다.	

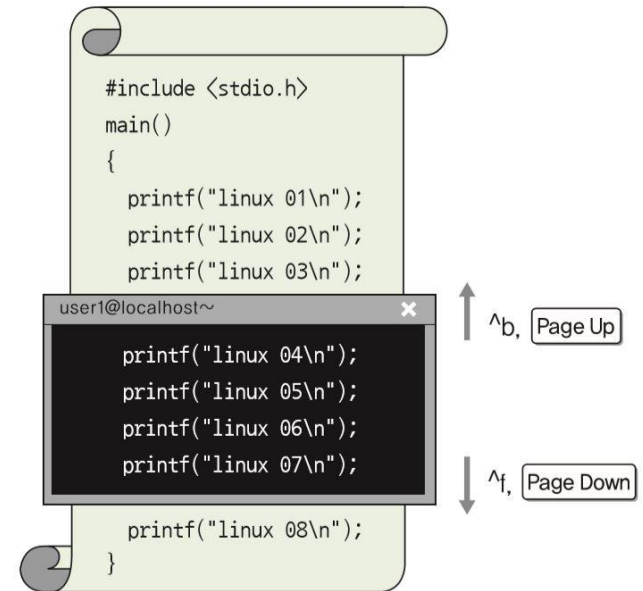


그림 3-4 화면 이동 명령키의 사용



02.vi 사용법

■ 화면 이동하기

■ 특정 행으로 바로 이동하기

- 원하는 행으로 커서를 바로 이동
- 예시)
 - 50G → 50번째 행으로 이동
 - :30[Enter] → 30번째 행으로 이동
 - 행 번호 대신 \$를 입력하면 파일의 마지막 행으로 이동

표 3-8 특정 행으로 바로 이동하는 명령키


명령키	기능
G( +g)	파일의 마지막 행으로 커서가 이동한다.
행 번호G( +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 수정

- 수정을 마치면 [esc] 를 눌러서 명령 모드로 돌아가야 수정이 완료

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 들면 3cw는 커서 위치부터 세 단어를 수정한다.
s, #s	커서 위치부터  를 입력할 때까지 수정한다. #에는 수정할 글자의 수를 지정한다. 예를 들면 5s는 커서 위치부터 다섯 글자를 수정한다.
cc	커서가 위치한 행의 내용을 모두 수정한다.
C	커서 위치부터 행의 끝까지 수정한다.

02.vi 사용법

■ 내용 수정

■ 한 글자 수정하기: r 명령 키

- 수정하려는 글자 위에 커서를 놓은 후 r 명령 키를 먼저 입력하고 바꾸려는 새 글자를 입력
- r 명령 키는 계속 명령 모드를 유지하므로 수정한 후 Esc키를 누를 필요가 없음

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

■ 단어 수정하기: cw, #s 명령 키

- 한 단어를 바꿀 때는 cw 명령 키나 #s 명령 키를 사용
- vi에서는 단어를 공백문자나 특수문자로 구분
 - ex) 'winux'를 다른 단어로 바꾸려면 cw 명령 키나, 글자 수가 다섯 개이므로 5s로 수정

```
CentOS study            → cw 명령키 입력 시 winux가 사라진다.  
I like linu CentOS linux  
  
~  
(생략)  
~  
— 끼워넣기 —
```

```
CentOS editor study     → 수정 완료 후 [Esc]를 눌러야 명령 모드로 전환된다.  
I like linu CentOS linux  
  
~
```

02.vi 사용법

■ 내용 수정

■ 행 단위 수정하기: C, cc 명령키

- C 명령키
 - 현재 커서가 위치한 r부터 행의 끝까지 수정하려면 C 명령키를 사용
 - r부터 행 의 마지막까지 모두 지워지고 입력 모드로 전환되어 입력을 기다림

CentOS edito **C** → r부터 모두 지워지고 입력 모드로 전환된다.

I like linu CentOS linux

~

(생략)

~

— 끼워넣기 —

- 수정할 대상의 글자 수와 상관없이 원하는 대로 입력

CentOS editor vi

I like linu CentOS linux

~

02.vi 사용법

■ 내용 수정

■ 행 단위 수정하기: C, cc 명령키

- cc 명령키

- cc 명령키를 입력하면 현재 행의 모든 내용이 삭제되고, 커서가 행의 처음으로 이동하여 새로운 입력을 기다림

```
I like linu CentOS linux
~
(생략)
~
— 끼워넣기 —
```

→ 모두 지워지고 행의 처음으로 이동한다. 입력 모드로 전환된다.

- 원하는 내용을 입력하고 [esc] 를 눌러 명령 모드로 전환하면 수정이 완료

```
CentOS editor vi study
I like linu CentOS linux
~
```


02.vi 사용법

■ 내용 삭제

- 입력 모드에서 내용을 입력하는 도중에 틀린 글자를 삭제할 때는 [←] 키나 [Delete]를 사용
- 명령 모드에서 글자나 행을 삭제하려면 [표 3-10]과 같은 명령키를 사용
- 한 번에 여러 글자나 행을 삭제하려면 명령키 앞에 글자나 행의 수를 숫자로 지정
 - ex) 한 글자를 삭제하는 것은 x 명령키, 세 글자를 삭제하려면 3x

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.

- x 명령 키로 현재 커서가 놓인 'l' 한 글자만 삭제

```
CentOS editor vi study
I like llinu CentOS linux
~
```

```
CentOS editor vi study
I like llinu CentOS linux
~
```

→ x 명령키로 커서 위치의 l을 삭제한다.

02.vi 사용법

■ 내용 삭제

- 나머지 글자를 모두 지우려면 글자 수를 지정하여 3x를 입력하거나 단어를 지우는 dw 명령키를 사용
- 3x로 지우는 경우와 dw로 지우는 경우, 지워지고 난 후 커서 의 위치에 차이가 있음

```
CentOS editor vi study
I like CentOS linux      → dw로 삭제하면 커서가 다음 단어의 첫 글자로 이동한다.

~
```

```
CentOS editor vi study
I like CentOS linux      → 3x로 삭제하면 커서가 공백문자에 위치한다.

~
```

- 현재 커서 위치부터 행의 끝까지 삭제하려면 D([Shift]+d) 명령키를 입력

```
CentOS editor vi study
I like

~
```

- 커서가 위치한 현재 행을 지우려면 dd 명령키를 입력

```
CentOS editor vi study


~
```

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.

- u 명령키를 입력하면 앞의 예에서 삭제되었던 행이 복구

```
CentOS editor vi study
I like CentOS linux
~
```

- 모든 작업을 취소하고 원래대로 복구하려면 U 명령키를 이용

02.vi 사용법

■ [따라해보기] vi로 입력·수정·삭제·복구하기

- ① vi로 새로운 파일인 exec.txt 파일을 실행

```
[user1@localhost ch3]$ vi exec.txt
```

- ② i를 입력하여 입력 모드로 전환하고 다음 내용을 입력

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

- ③ [esc]를 입력하여 명령 모드로 전환하고 :w 명령으로 파일 내용을 저장

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

02.vi 사용법

■ [따라해보기] vi로 입력·수정·삭제·복구하기

- ④ 커서를 3행의 beginner로 이동
3G와 | 명령키(또는 오른쪽 화살표나 w 명령키와 h 명령키)를 사용하여 이동할 수 있음

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

- ⑤ beginner를 expert로 수정cw나 8s 명령키를 사용한다. 수정 후에는[esc]를 눌러 명령 모드로 전환

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

- ⑥ - 명령키로 2행의 첫 글자인 N으로 이동

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

02.vi 사용법

■ [따라해보기] vi로 입력·수정·삭제·복구하기

- ⑦ w 명령키로 커서를 meet로 이동하고, 단어 meet를 dw 명령키로 삭제

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

- ⑧ 단어 y부터 행의 끝까지 D([Shift]+d) 명령키로 삭제

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

02.vi 사용법

■ [따라해보기] vi로 입력·수정·삭제·복구하기

- ⑨ U 명령키로 2행에서 한 모든 삭제를 취소

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

- ⑩ :wq 명령으로 파일 내용을 저장하고 종료

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

02.vi 사용법

■ 복사 및 잘라 붙이기

■ 복사하기, 잘라내기, 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.

- yy 명령키로 해당 행을 복사하고, 원하는 행으로 커서를 이동하여 p 명령 키를 입력하면 '복사하여 붙이기'
- dd 명령키로 행을 삭제한 후 p 명령키를 입력하면 '잘라서 붙이기'
- 복사하거나 잘라내기를 하면 내용이 임시 버퍼에 저장되기 때문에 복사하거나 잘라내기를 한 뒤에는 다른 명령을 사용하지 말고 즉시 원하는 위치로 이동하여 붙이기를 하는 것이 좋음

02.vi 사용법

■ 복사 및 잘라 붙이기

■ 복사하기, 잘라내기, 붙이기

- 앞서 저장해둔 test.txt 파일을 vi로 열면 커서는 f 위에 위치

```
CentOS editor vi study
I like  CentOS linux
~
```

- 현재 커서가 있는 행만 복사하려면 그냥 yy 명령 키만 입력
- 1행과 2행을 함께 복사하려면 2yy를 입력
- ex) 2yy 명령 키로 두 행을 모두 복사한 다음 아랫행으로 이동하여 p 명령 키로 붙이기

```
CentOS editor vi study
I like  CentOS linux
CentOS editor vi study
I like  CentOS linux
~
```

- dd 명령 키는 삭제뿐만 아니라 잘라내기를 할 때도 사용
- ex) 현재 위치인 3행을 잘라 2행 위에 붙이기 → dd 키를 입력한 다음 커서를 2행으로 이동하여 p 키를 입력

```
CentOS editor vi study
CentOS editor vi study
I like  CentOS linux
I like  CentOS linux
~
```

02.vi 사용법

■ 복사 및 잘라 붙이기

■ 네임드 버퍼 사용하기

- 언네임드 버퍼
 - 이름을 붙이지 않은 버퍼
 - yy 명령 키로 복사하거나 dd 명령 키로 잘라낼 경우, 복사하거나 잘라낸 내용이 저장
 - 언네임드 버퍼는 내용을 하나만 저장
- 네임드 버퍼
 - 이름을 붙여서 사용할 수 있는 버퍼
 - 네임드 버퍼를 사용하면 각각 다른 이름을 붙인 버퍼에 독립적으로 내용을 저장하고 사용
 - 네임드 버퍼에 이름을 붙일 때는 "+문자" 또는 "+숫자(숫자 버퍼)"의 형태로 사용
 - 네임드 버퍼 : "a", "b", "c", "d, ..., "z
 - 숫자 버퍼 : "1, "2, ..., "9
 - 네임드 버퍼에 커서가 위치한 행을 저장하려면 '버퍼 이름+yy'를 입력
 - "a 버퍼에 복사한 내용을 저장하려면 "ayy를 입력
 - 네임드 버퍼의 내용을 붙이려면 "ap와 같이 '버퍼 이름+p'를 입력

02.vi 사용법

■ 복사 및 잘라 붙이기

■ 네임드 버퍼 사용하기

- 현재 파일 내용

```
CentOS editor vi study
CentOS editor vi study
I like CentOS linux
I like CentOS linux
~
```

- 2행을 "a 버퍼에 잘라서 저장하고, "a를 먼저 입력 후 dd 입력

```
CentOS editor vi study
I like CentOS linux
I like CentOS linux
~
```

- 다시 2행을 잘라 "b 버퍼에 저장하고

- 커서를 1행으로 이동
- "b 버퍼의 내용을 1행 윗 행에 붙여 넣으려면 "bP를 입력
- "a 버퍼의 내용을 1행 아랫 행에 붙여 넣으려면 "aP를 입력

```
I like CentOS linux
CentOS editor vi study
CentOS editor vi study
I like CentOS linux
~
```

02.vi 사용법

■ 복사 및 잘라 붙이기

■ 네임드 버퍼 사용하기

- 현재 파일 내용

```
CentOS editor vi study
CentOS editor vi study
I like  CentOS linux
I like  CentOS linux
~
```

- 2행을 "a 버퍼에 잘라서 저장하고, "a를 먼저 입력 후 dd 입력

```
CentOS editor vi study
I like  CentOS linux
I like  CentOS linux
~
```

- 다시 2행을 잘라 "b 버퍼에 저장하고

- 커서를 1행으로 이동
- "b 버퍼의 내용을 1행 윗 행에 붙여 넣으려면 "bP를 입력
- "a 버퍼의 내용을 1행 아랫 행에 붙여 넣으려면 "aP를 입력

```
I like  CentOS linux
CentOS editor vi study
CentOS editor vi study
I like  CentOS linux
~
```

02.vi 사용법

■ 복사 및 잘라 붙이기

- 마지막 행 모드에서 복사하기, 잘라내기, 붙이기
 - 범위 지정하기
 - .(마침표)는 커서가 위치한 현재 행을, \$는 마지막 행을, %는 전체 행을 의미

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.

- 복사하기, 잘라내기, 붙이기

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

명령키	기능
:#y	#로 지정한 행을 복사한다. 예를 들면 3y는 세 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 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

■ 복사 및 잘라 붙이기

■ 마지막 행 모드에서 복사하기, 잘라내기, 붙이기

- 복사하기, 잘라내기, 붙이기
 - 커서는 2행에 위치

```
I like  CentOS linux
CentOS editor vi study
CentOS editor vi study
I like  CentOS linux
~
```

- :2,3d 명령키로 2행과 3행을 잘라내고 :2pu 명령키로 2행 다음에 잘라낸 것을 붙임

```
I like  CentOS linux
I like  CentOS linux
CentOS editor vi study
CentOS editor vi study
~
```

- :1,2y 명령키로 1행부터 2행까지 복사하고 :4pu 명령키로 붙여넣음

```
I like  CentOS linux
I like  CentOS linux
CentOS editor vi study
CentOS editor vi study
I like  CentOS linux
I like  CentOS linux
~
```

02.vi 사용법

■ 검색 및 바꾸기

■ 검색하기

- 검색하기 위해 마지막 행으로 이동할 때는 :이 아니라 /이나 ?를 입력
- /이나 ? 다음에 찾고자 하는 문자열을 입력하고
- [Enter]를 누르면 검색이 진행되고, 해당 문자열을 찾으면 그 문자열의 시작 위치로 커서가 이동

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

■ 검색 및 바꾸기

■ 검색하기

- 커서가 6행에 있는 상태에서 문자열을 검색하기 위해 /을 입력하면 커서가 마지막 행으로 이동
- 검색할 문자열인 'CentOS'를 입력하고 [Enter]를 누르면 커서 위치보다 뒤쪽에 위치한 같은 행의 CentOS로 커서가 이동

```
I like CentOS linux
I like CentOS linux
CentOS editor vi study
CentOS editor vi study
I like CentOS linux
I like CentOS linux
~
(생략)
/CentOS
```

- 계속 CentOS를 검색하려고 n을 입력하면 6행이 파일의 마지막 행이므로 '끝까지 찾았음, 처음부터 계속'이라는 메시지를 출력하고 1행의 CentOS로 커서가 이동

```
I like CentOS linux
I like CentOS linux
CentOS editor vi study
CentOS editor vi study
I like CentOS linux
I like CentOS linux
~
(생략)
끝까지 찾았음, 처음부터 계속
```


02.vi 사용법

■ 검색 및 바꾸기

■ 바꾸기

- 기존의 문자열을 다른 문자열로 바꾸려면 먼저 :을 입력하여 마지막 행 모드로 전환
- 커서 위치의 문자열만 바꿀 수도 있고, 파일 전체나 특정 범위 내에서 해당하는 문자열을 모두 바꿀 수도 있음

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

■ 검색 및 바꾸기

■ 바꾸기

- ex) 현재 커서의 CentOS를 CENTOS로 변경할 경우 명령은 :s/CentOS/CENTOS/

```
I like  CentOS linux
I like  CentOS linux
CentOS editor vi study
CentOS editor vi study
I like  CentOS linux
I like  CentOS linux
~
(생략)
:s/CentOS/CENTOS/
```

- ex) 3~4행에 있는 editor를 CentOS로 바꿀 경우 의 명령은 :3,4s/editor/CentOS/

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

02.vi 사용법

■ 검색 및 바꾸기

■ 바꾸기

- 명령에 g가 있을 경우 해당 행이나 범위 내에서 바꾸려는 문자열을 모두 찾아 다른 문자열로 바꿈
 - ex) 행에서 :s/fedora/FEDORA/을 수행하고 4행에서 :s/fedora/FEDORA/g를 수행하여 결과의 차이를 비교

```
I like  CentOS linux
I like  CentOS linux
CENTOS CentOS vi study
CENTOS CentOS vi study
I like  CentOS linux
I like  CentOS linux
~
(생략)
:s/CentOS/CENTOS/g
```

- ex) 파일 전체의 문자열을 바꿀 경우 :%s/CentOS/CENTOS/g 또는 :1,\$s/CentOS/CENTOS/g를 사용

```
I like  CentOS linux
I like  CentOS linux
CENTOS CentOS vi study
CENTOS CentOS vi study
I like  CentOS linux
I like  CentOS linux
~
(생략)
4개 바꿨음 4 행에서
```

02.vi 사용법

■ 검색 및 바꾸기

■ 바꾸기

- 명령에 g가 있을 경우 해당 행이나 범위 내에서 바꾸려는 문자열을 모두 찾아 다른 문자열로 바꿈
 - ex) 행에서 :s/fedora/FEDORA/을 수행하고 4행에서 :s/fedora/FEDORA/g를 수행하여 결과의 차이를 비교

```
I like  CentOS linux
I like  CentOS linux
CENTOS CentOS vi study
CENTOS CentOS vi study
I like  CentOS linux
I like  CentOS linux
~
(생략)
:s/CentOS/CENTOS/g
```

- ex) 파일 전체의 문자열을 바꿀 경우 :%s/CentOS/CENTOS/g 또는 :1,\$s/CentOS/CENTOS/g를 사용

```
I like  CentOS linux
I like  CentOS linux
CENTOS CentOS vi study
CENTOS CentOS vi study
I like  CentOS linux
I like  CentOS linux
~
(생략)
4개 바꿨음 4 행에서
```

02.vi 사용법

■ [따라해보기] vi 편집 방법 익히기

- ① vi로 새로운 파일인 exec2.txt 파일을 실행

```
[user1@localhost ch3]$ vi exec2.txt
```

- ② i를 입력하여 입력 모드로 전환하고 다음 내용을 입력

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

- ③ [esc]를 입력하여 명령 모드로 전환하고 :w 명령으로 파일 내용을 저장

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

- ④ 1행을 복사하여 3행 다음에 붙임

```
Good morning everyone.  
My name is Gildong Hong.  
This is a living room.  
Good morning everyone. → ① 1G 입력 → ② yy 입력 → ③ jj 입력 → ④ p 입력  
~  
~
```

02.vi 사용법

■ [따라해보기] vi 편집 방법 익히기

⑤ 파일에서 morning이란 단어를 검색

```
Good morning everyone.  
My name is Gildong Hong.  
This is a living room.  
Good morning everyone. → ① /morning [Enter] 입력 → ② n 입력  
~  
/morning
```

⑥ 파일에서 morning이란 단어를 모두 찾아 afternoon으로 바꿈

```
Good afternoon everyone.  
My name is Gildong Hong.  
This is a living room.  
Good afternoon everyone. → :%s/morning/afternoon/g [Enter] 입력  
~  
:%s/morning/afternoon/g
```

⑦ 저장한 후 종료

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

02.vi 사용법

■ 기타 유용한 명령키

- 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

■ 기타 유용한 명령키

■ 파일 읽어오기, 여러 파일 편집하기

- 다른 파일 읽어오기
 - test.txt 파일을 열고 커서를 4행으로 이동

```
I like  CENTOS linux
I like  CENTOS linux
CENTOS CENTOS vi study
CENTOS CENTOS vi study
I like  CENTOS linux
I like  CENTOS linux
~
```

- :r exec2.txt를 실행하면 exec2.txt 파일의 내용이 test.txt 파일의 4행 다음에 삽입

```
I like  CENTOS linux
I like  CENTOS linux
CENTOS CENTOS vi study
CENTOS CENTOS 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  CENTOS linux
I like  CENTOS linux
~
```


02.vi 사용법

■ 기타 유용한 명령키

- 파일 읽어오기, 여러 파일 편집하기
 - 파일 편집을 마치고 다른 파일 편집하기
 - :e 명령 키는 현재 작업 중인 파일의 작업을 마치고 다른 파일을 편집하려고 할 때 사용
 - ex) test.txt 파일 편집을 완료하고 exec.txt 파일 편집으로 바꾸려면 :e exec.txt 사용
 - 이때 작업 중이던 파일을 먼저 저장하고 :e 명령을 실행해야 함
 - 파일을 저장하지 않고 :e exec.txt를 실행하면 다음과 같은 오류 메시지가 출력

```
I like  CENTOS linux
I like  CENTOS linux
CENTOS CENTOS vi study
CENTOS CENTOS vi study
Good afternoon everyone.
This is a living room.
(생략)
~
E37: 마지막으로 고친 뒤 저장되지 않았습니다 (무시하려면 ! 더하기)
계속하려면 엔터 혹은 명령을 입력하십시오
```

- 여러 파일 편집하기

```
[user1@localhost ch3]$ vi test.txt exec.txt exec2.txt
```

- vi를 시작할 때 파일명을 여러 개 지정
- 파일 작업을 마치고 다음 파일로 이동하려면 :n을 입력

02.vi 사용법

■ 기타 유용한 명령키

■ vi에서 셸 명령 사용하기

표 3-18 셸 명령 실행 명령키

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

- `:!` 명령키 이용하기
 - vi를 빠져나가거나 하는 번거로움 없이 바로 이용할 수 있다는 장점
 - 다시 vi 작업으로 돌아가려면 `Enter`키 입력
 - ex) test.txt 파일을 편집하는 중에 `:! ls` 명령 실행

(생략)

~

~

`:! ls`

```
[user1@localhost ch3]$ vi test.txt
```

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

계속하려면 엔터 혹은 명령을 입력하십시오

02.vi 사용법

■ 기타 유용한 명령키

■ vi에서 셸 명령 사용하기

- :sh 명령키 이용하기
 - '! 셸 명령'은 한 번에 하나의 셸 명령만 실행
 - 실행할 셸 명령이 여러 개라면 :sh 명령 키로 vi를 잠시 빠져나가서 셸 작업을 수행하고 다시 돌아오는 것이 편리

(생략)

~

~

:sh

- ex) 필요한 셸 작업을 마치고 다시 vi로 돌아오려면 exit를 입력

```
[user1@localhost ch3]$ ls
```

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

```
[user1@localhost ch3]$ exit
```

02.vi 사용법

■ 기타 유용한 명령키

■ 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 아랫 행을 연결하여 한 행으로 만든다.
.(마침표)	바로 직전에 했던 명령을 반복한다.
~(틸드)	커서 위치의 글자를 대문자나 소문자로 바꾼다.

- 화면 다시 출력하기

- vi 작업 도중에 시스템 메시지나 다른 사용자가 보낸 메시지가 출력되어 화면이 이상하게 보일 때 **Ctrl** + l (소문자 L) 명령키를 입력하면 메시지가 사라지고 원래 작업 중이던 내용이 다시 출력

02.vi 사용법

■ 기타 유용한 명령키

■ 기타 명령키

- 행 연결하기
 - 행을 연결할 때는 J 명령키를 사용
 - 커서가 위치한 행과 다음 행을 하나의 행으로 만들어줌

```
I like  CENTOS linux
I like  CENTOS linux
CENTOS CENTOS vi study
(생략)
~
```

- J 명령키를 입력하면 두 번째 행과 합쳐짐
- 두 행이 합쳐질 때 사이에 공백문자가 삽입되며, 커서가 그 공백문자로 이동

```
I like  CENTOS linux I like  CENTOS linux
CENTOS CENTOS vi study
(생략)
~
```

02.vi 사용법

■ 기타 유용한 명령키

■ 기타 명령키

- 이전 명령 반복하기
 - .(마침표) 명령키는 바로 앞에 했던 명령을 반복적으로 수행
 - 커서 이동 키는 . 명령키를 입력해도 반복 수행되지 않음

```
I like  CENTOS linux I like  CENTOS linuxCENTOS CENTOS vi study
(생략)
~
```

- 대·소문자 전환하기
 - ~ 명령키는 커서가 놓인 곳의 글자가 소문자이면 대문자로, 대문자이면 소문자로 바꿔줌
 - 앞의 예에서 커서를 x로 옮기고 ~ 명령키를 입력하면 X로 바뀌고 커서가 오른쪽으로 한 칸 이동

```
I like  CENTOS linux I like  CENTOS linuXCENTOS CENTOS vi study
(생략)
~
```

02.vi 사용법

■ [따라해보기] 기타 유용한 명령 익히기

- ① vi로 새로운 파일인 exec3.txt 파일을 옴

```
[user1@localhost ch3]$ vi exec3.txt
```

- ② i를 입력하여 입력 모드로 전환하고 다음 내용을 입력

```
Good  
morning  
everyone.  
~  
:w
```

- ③ [esc]를 입력하여 명령 모드로 전환하고 :w 명령으로 파일 내용을 저장

```
Good  
morning  
everyone.  
~
```

02.vi 사용법

■ [따라해보기] 기타 유용한 명령 익히기

- ④ 3행 다음에 exec2.txt 파일을 읽어들이м

```
Good
morning
everyone.
Good afternoon everyone. → :r exec2.txt [Enter]를 입력한다.
This is a living room.
My name is Gildong Park.
This is a living room.
My name is Gildong Hong.
Good afternoon everyone.
~
"exec2.txt" 6L, 138C
```

- ⑤ 1행과 2행을 연결하여 한 행으로 만듦

```
Goodmorning → ① 1G로 이동 → ② J 입력
everyone.
Good afternoon everyone.
(생략)
~
"exec2.txt" 6L, 138C
```


02.vi 사용법

■ [따라해보기] 기타 유용한 명령 익히기

- ⑦ 반복 명령키로 1행과 2행을 연결하여 한 행으로 만들

```
Good morning everyone. → .을 입력한다.  
Good afternoon everyone.  
(생략)  
~  
"exec2.txt" 6L, 138C
```

- ⑧ ~명령키로 1행의 everyone을 EVERYONE으로 변경

```
Good morning EVERYONE. → ~~~~~ 입력  
Good afternoon everyone.  
(생략)  
~  
"exec2.txt" 6L, 138C
```

02.vi 사용법

■ [따라해보기] 기타 유용한 명령 익히기

- ⑧ exec3.txt 파일을 저장하고, 셸 명령을 실행하여 현재 디렉터리의 경로를 확인

```
Good morning EVERYONE.  
Good afternoon everyone.  
(생략)  
~
```

```
:! pwd → ① :w 로 저장 → ② :! pwd 
```

```
[user1@localhost ch3]$ vi exec3.txt
```

```
/home/user1/linux_ex/ch3 → ① 경로 정보를 출력한다.
```

```
계속하려면 엔터 혹은 명령을 입력하십시오 → ② 를 입력하여 vi로 복귀한다.
```

- ⑨ vi를 종료

03.vi 환경 설정

■ vi 환경 설정

- vi의 환경을 설정하는 세 가지 방법
 - 사용자 홈 디렉터리에 .exrc 파일로 저장
 - 환경 변수 EXTINIT에 지정
 - vi의 마지막 행 모드에서 명령으로 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 nu를 입력하면 다음과 같이 행 번호가 표시
- 행 번호는 사용자의 편의를 위해 보이는 것으로 파일에 저장되지는 않음
- :set nonu를 입력하면 행 번호가 없어짐

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

■ 특수문자 표시

- 특수문자를 보려면 :set list 명령을 입력
- exec2.txt 파일에 탭을 하나 추가하고 :set list 명령을 실행

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 ^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 명령은 현재 사용 중인 vi의 환경 설정 상태를 보여줌

```
~
:set
— 옵션 —
commentstring=      incsearch          scrolloff=5          ttymouse=xterm2
display=truncate    langnoremap        showcmd              viminfo='20,"50
filetype=text       nolangremap          syntax=text          wildmenu
helplang=ko          nrformats=bin,hex    ttimeout              t_Sb=^[[4%dm
history=200           ruler                ttimeoutlen=100      t_Sf=^[[3%dm
hlsearch             scroll=11             ttyfast
backspace=indent,eol,start
comments=fb:~,fb:*
fileencoding=utf-8
fileencodings=ucs-bom,utf-8,latin1
guicursor=n-v-c:block,o:hor50,i-ci:hor15,r-cr:hor30,sm:block,a:blinkon0
계속하려면 엔터 혹은 명령을 입력하십시오
```

03.vi 환경 설정

■ 모든 환경 변수 표시

- vi의 환경 설정에 사용되는 모든 환경 변수와 현재 설정 값을 보여주는 명령은 :set all

```
:set all
— 옵션 —
aleph=224          fileformat=unix      mouse=            swapfile
noarabic           nofileignorecase    mousemodel=extend swapsync=fsync
arabicshape       filetype=text       mousetime=500    switchbuf=
noallowrevins      fixendofline        nrformats=bin,hex synmaxcol=3000
noaltkeymap        nofkmap             nonumber          syntax=text
ambiwidth=single   foldclose=          numberwidth=4     tabline=
noautochdir        foldcolumn=0        omnifunc=         tabpagemax=10
noautoindent       foldenable          operatorfunc=     tabstop=8
noautoread         foldexpr=0          nopaste           tagbsearch
noautowrite        foldignore=#        pastetoggle=      tagcase=followic
noautowriteall     foldlevel=0         patchexpr=        taglength=0
background=light   foldlevelstart=-1   patchmode=        tagrelative

nobackup           foldmethod=manual   nopreserveindent  tagstack
backupcopy=auto    foldminlines=1      previewheight=12  term=xterm
backupext=~        foldnestmax=20      nopreviewwindow   notermbidi
backupskip=/tmp/*  formatexpr=         printdevice=      termencoding=
balloondelay=600   formatoptions=tcq   printencoding=    notermguicolors
noballoonevalterm  formatprg=          printfont=courier termkey=
balloonexpr=       fsync               printmbcharset=   termsize=
belloff=          nogdefault          printmbfont=      termwinkey=
nobinary           helphheight=20      printoptions=     termwinsize=
— 더 —
```

03.vi 환경 설정

■ 환경 설정 파일과 변수 이용

- .exrc 파일에 환경 설정
 - 사용자 홈 디렉터리에 .exrc 파일로 저장
 - 기본적으로 없는 파일이므로 사용자가 만들어야 함
 - 파일에는 set 명령과 옵션만 지정
 - 이 파일이 있을 경우 vi를 시작할 때마다 확인하므로 모든 파일에 동일하게 적용 가능

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

■ EXINIT 환경 변수에 설정하기

- vi 환경 설정은 다음과 같이 셸의 환경 변수인 EXINIT에도 가능

```
[user1@localhost ch3]$ EXINIT='set nu list'
[user1@localhost ch3]$ export EXINIT
[user1@localhost ch3]$
```

Thank you
