vi(Visual) 편집기

UNIX 텍스트 편집기 종류 p158

- **vi**
  - 대부분의 **UNIX** 시스템의 기본 텍스트 에디터
- **ed**
  - 초기 **UNIX** 시스템용 텍스트 에디터
  - 텍스트 에디터의 시초
- **emacs**
  - 범용 **UNIX** 텍스트 에디터

vi 편집기

- 작성하는 파일의 내용을 화면 전체에 보여주는 텍스트 에디터
- 여백 머리말 넣기 등과 같은 워드 프로세서의 기능은 제공하지 않음
- 작업 버퍼 안에서 모든 작업을 수행
  - 사용자가 작업 버퍼의 내용을 파일로 옮기기 전까지 작업한 내용을 파일에 저장하지 않음
- 두개의 모드
  - 명령 모드 : 사용자가 입력하는 것을 명령어로 인식
  - 입력 모드 : 사용자가 입력하는 문자를 텍스트로 여겨 이를 화면에 보여 줌

vi 시작과 종료

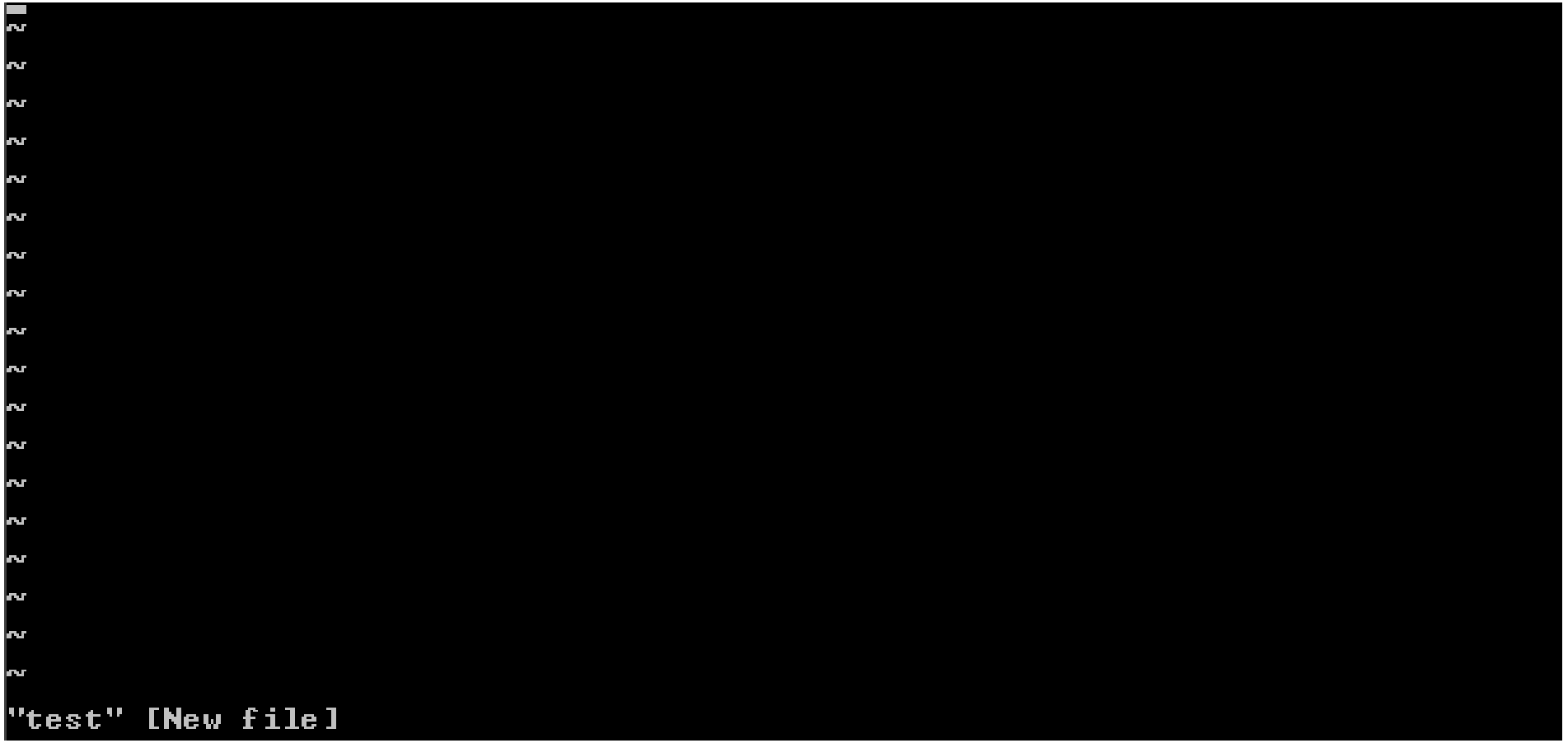
vi 시작

```
$ vi [옵션] 파일명
```

옵 션	설 명
-R	읽기 전용(read only)으로 파일을 연다
-$	파일의 맨 마지막 라인으로 커서를 위치시키면서 파일을 연다
-*n*	*n*으로 지정한 라인에 커서를 위치시키면서 파일을 연다
-r	손실된 파일을 복구한다

vi 시작 (계속)

```
$ vi test
```



vi 편집 모드

- 명령 모드 (command mode)
  - vi를 실행했을 때의 초기 모드
  - 사용자가 입력하는 값은 화면에 나타나지 않음
  - 사용자가 입력한 값을 명령어로 인식

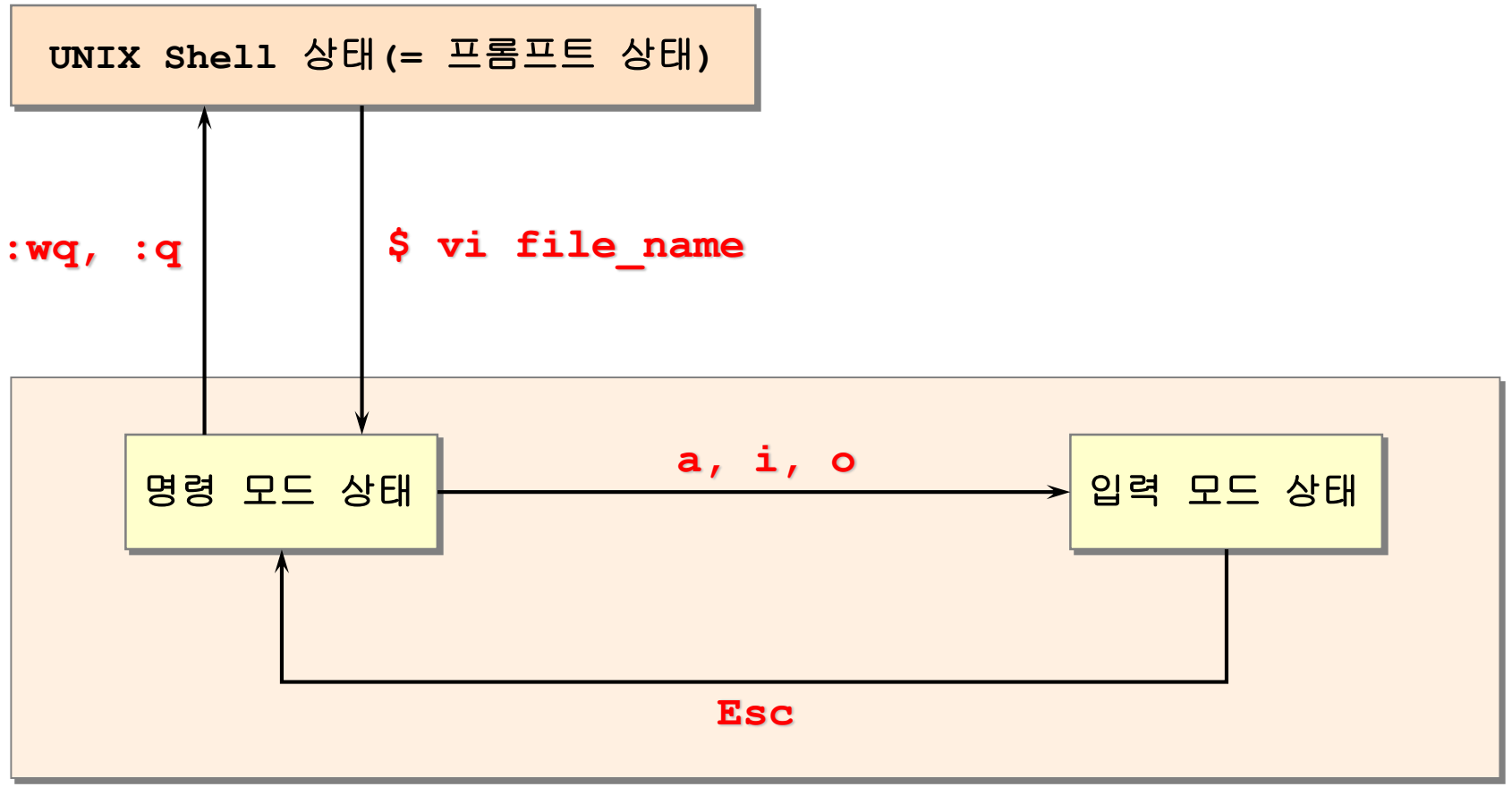
명령어	설 명
a(append)	현재 커서가 있는 위치 다음부터 입력하는 기능
i(insert)	현재 커서가 있는 위치에 새로운 텍스트를 왼쪽에 삽입하는 기능
o(open)	현재 커서가 있는 다음 행에 삽입하여 편집하는 기능

vi 편집 모드 (계속)

- 입력 모드 (input mode)
  - 데이터를 입력하기 위한 모드
  - 사용자가 입력하는 문자를 텍스트로 여겨 이를 화면에 보여 줌
- 마지막 라인 모드
  - 명령 모드에서 콜론(:) 입력
  - 마지막 라인 모드의 기능
    - 파일 저장 및 파일 읽기 등의 파일 관리 기능
    - 일시적으로 쉘 명령을 사용할 수 있는 기능
    - 파일 내의 특정 문자열 같은 패턴을 검색하는 기능
    - 여러가지 옵션을 사용하여 vi 환경을 편리하게 해주는 기능
    - 매크로(macro) 사용 기능

vi 편집 모드 (계속)

- 편집 모드 변환



vi 편집 모드 (계속)

- 작업중 주기적 저장 (**:w, :w!**)
  - 작업 버퍼(**work buffer**)의 내용을 현재 파일에 저장
  - **:w**와 **:w!**의 차이점

:w 파일명	동일한 파일명 존재시 저장되지 않음
:w! 파일명	동일한 파일명 존재시 덮어 씀
:w! >> 파일명	현재 작업 버퍼의 내용을 기존 파일에 추가 저장

- 화면 깨끗이 하기 (**<Ctrl+L>**)
  - 작업 화면이 지저분할 때, 명령 모드에서 **<Ctrl+L>**키를 사용하여 화면을 깨끗하게 재구성

vi 편집 모드 (계속)

- 일시적 쉘 사용하기 (:!, :sh)
  - :!(쉘 명령어)
    - 명령 모드에서 :! 다음에 필요한 쉘 명령어 입력

```
~
:!who
root         console      Sep  2 15:29
root         :0           Sep  2 15:29
root         pts/1        Sep  2 15:29
root         pts/2        Sep  2 15:38
root         pts/3        Sep  7 15:06
admin        pts/4        Sep 14 05:14
s0110370     pts/72       Sep 10 09:15
[Hit return to continue]
```

vi 편집 모드 (계속)

- 일시적 쉘 사용하기 (**:!, :sh**)
  - **:sh**
    - 명령 모드에서 쉘을 호출하는 **:sh** 명령 입력
    - **vi**로 돌아가려면 **exit**

```
~
:sh
$ who
root         console      Sep  2 15:29
root         :0           Sep  2 15:29
root         pts/1        Sep  2 15:29
root         pts/2        Sep  2 15:38
root         pts/3        Sep  7 15:06
admin        pts/4        Sep 14 05:14
s0110370     pts/72       Sep 10 09:15
$ exit
```

vi 종료

- 파일에 저장후 종료 (ZZ, :wq, :wq!)
  - :wq와 :wq!의 차이점

:wq 파일명	동일한 파일명 존재시 저장 및 종료되지 않음
:wq! 파일명	동일한 파일명 존재시 덮어쓰고 종료
:wq! >> 파일명	현재 작업 버퍼의 내용을 기존 파일에 추가 저장 및 종료

- 저장하지 않고 종료 (:q, :q!)
  - :q와 :q!의 차이점

:q!	현재 작업 버퍼의 내용을 저장하지 않고 무조건 vi 종료
:q	파일 내용 변경이 있으면 저장되지 않음

파일 편집 과정

텍스트 추가 (append)

명령어	설 명
a	현재 커서가 위치하는 다음부터 텍스트 추가
A	현재 라인의 가장 뒤쪽부터 텍스트 추가

텍스트 삽입 (insert)

명령어	설 명
i	현재 커서가 있는 위치에서 새로운 텍스트를 왼쪽으로 삽입
I	현재 라인의 가장 앞쪽부터 텍스트 삽입

빈라인 삽입 (open)

명령어	설 명
o	현재 라인 아래에 입력할 빈 라인 추가
O	현재 라인 위에 입력할 빈 라인 추가

삭제 명령 (delete)

명령어	설 명
x / *n*x	현재 커서 위치에서 한 문자(*n*문자)씩 삭제하고 커서는 오른쪽으로 이동
X / *n*X	현재 커서가 위치한 앞의 한 문자(*n*문자)씩을 삭제
dw / dW	현재 커서가 위치한 곳에서 뒤쪽의 한 단어씩 삭제
d0	현재 커서 위치에서 라인의 가장 앞쪽까지 모든 문자 삭제
D / d$	현재 커서 위치에서 라인의 가장 뒤쪽까지 모든 문자 삭제
dd / *n*dd	현재 커서가 위치한 한 라인(*n* 라인) 전체 삭제

:.,$d	현재 라인에서 작업 버퍼의 끝까지 모든 라인 삭제

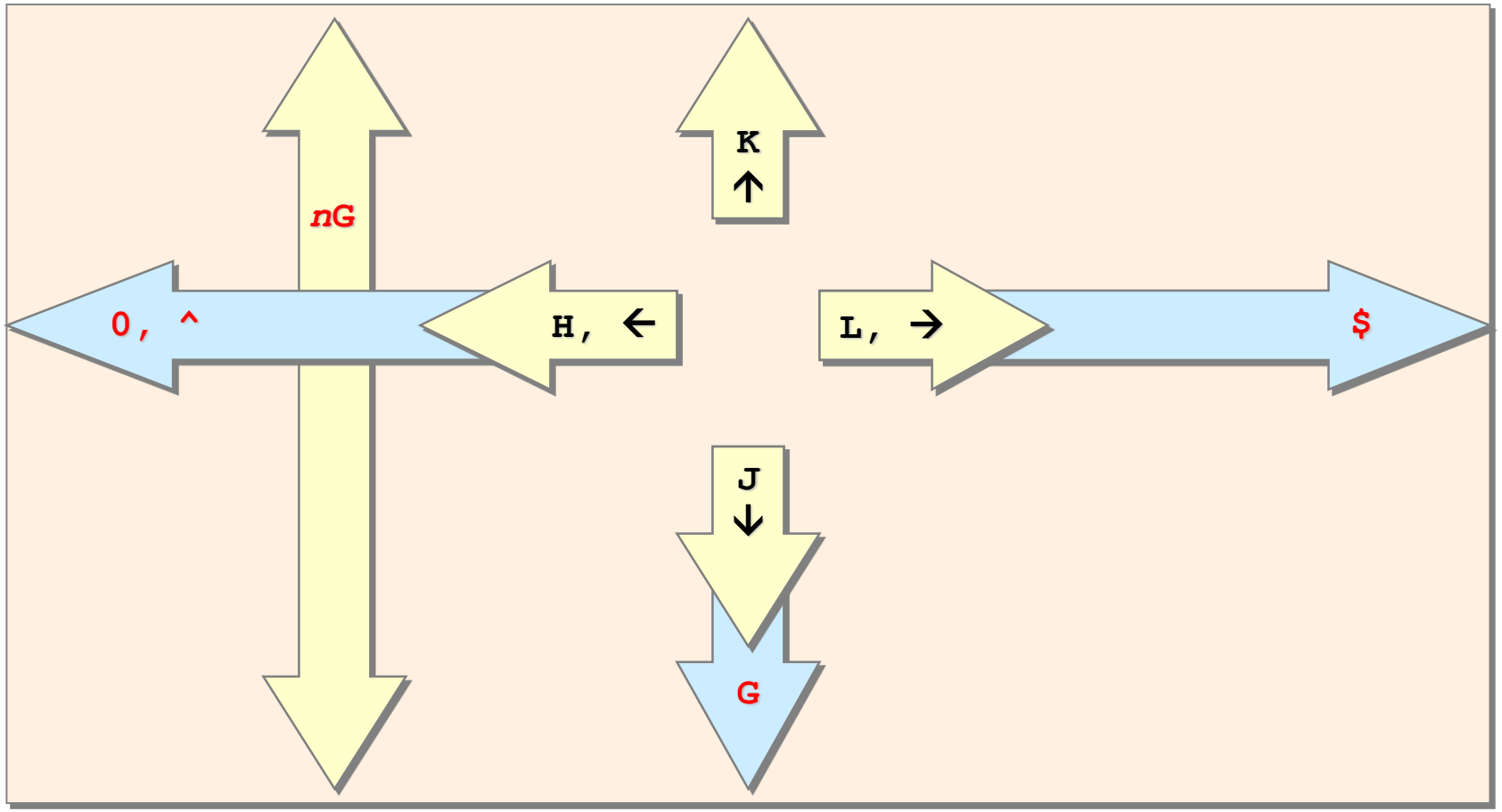
커서 이동 명령

명령어	설 명
L / →	오른쪽으로 한 문자씩 이동
H / ←	왼쪽으로 한 문자씩 이동
J / ↓	아래로 한 라인씩 이동
K / ↑	위로 한 라인씩 이동
$	라인의 마지막 문자로 커서 이동
0	라인의 첫번째 문자로 커서 이동
^	라인의 첫번째 비공백 문자로 커서 이동
*n*G	*n*번째 라인으로 커서 이동
G	마지막 라인으로 커서 이동

커서 이동 명령 (계속)

명령어	설 명
Ctrl+D	현재 화면의 아래 방향으로 ½ 이동
Ctrl+U	현재 화면의 위쪽 방향으로 ½ 이동
Ctrl+B	한 화면 전체(23라인)을 위쪽 방향으로 이동
Ctrl+F	한 화면 전체(23라인)을 아래 방향으로 이동

커서 이동 명령 (계속)



텍스트 교체

명령어	설 명
nr	- 현재 커서 위치에서 동일한 ***n***개의 문자 교체 - ***n***개의 동일한 문자 교체후 자동적으로 명령 모드로 전환 - **Esc** 키 필요 없음
R	- 여러 문자를 교체할 때 사용 - 현재 커서에서 **Esc** 누를 때까지 기존 텍스트 교체 - 라인의 끝에 도달하면 자동적으로 새로운 텍스트 추가하는 상태

텍스트 치환

명령어	설 명
***n*s**	- 현재 커서 위치에서 ***n***개의 문자 치환 - 치환될 문자의 끝 표시로서 ***n***번째 문자에 **$** 표시 - ***n***번째(**$**)에 도달하면 자동적으로 새로운 텍스트 추가하는 상태 - 치환 모드를 끝내려면 **Esc**
S	- 라인에 있는 모든 문자 치환 **(dd + O)**

텍스트 변경

명령어	설 명
cw	- 현재 커서 위치에서 단어 끝부분까지 변경 - 단어 끝에 끝 표시로 $ - 단어 끝($)에 도달하면 자동적으로 새로운 텍스트 추가하는 상태 - 변경 모드를 끝내려면 Esc
c$	- 현재 커서 위치에서 라인 끝 문자까지 변경 - 라인 끝에 끝 표시로 $ - 라인 끝($)에 도달하면 자동적으로 새로운 텍스트 추가하는 상태 - 변경 모드를 끝내려면 Esc
c^	- 현재 커서가 있는 라인의 처음 비공백 문자에서부터 커서가 있는 문자까지 변경 - 현재 커서에 끝 표시로 $ - $에 도달하면 자동적으로 새로운 텍스트 추가하는 상태 - 변경 모드를 끝내려면 Esc
cc	- S의 동일

텍스트 이동과 복사

- 텍스트 이동 (Delete-and-Put)

명령어	설 명
소문자 p	삭제되거나 복사된 라인이나 문자/단어들이 현재 라인이나 문자/단어 다음에 버퍼의 내용이 텍스트에 붙여짐
대문자 P	삭제되거나 복사된 라인이나 문자/단어들이 현재 라인이나 문자/단어 앞에 버퍼의 내용이 텍스트에 붙여짐

텍스트 이동과 복사 (계속)

- 텍스트 복사 **(Yank-and-Put)**

명령어	설 명
***n*yy** ***n*Y**	***n***개의 텍스트 라인 복사 저장
***n*yw** ***n*yW**	***n***개의 단어 복사 저장
y0	커서의 왼쪽부터 라인의 처음까지 텍스트 복사 저장
y$	현재 커서 위치에서 라인의 끝까지 텍스트 복사 저장

그 외의 유용한 vi 편집 명령

- 2개의 분리된 라인 결합 - **J**
- 마지막 실행했던 명령어 반복 - **.(dot)**
- 마지막 실행했던 명령어 취소
  - **u** : 마지막 실행 명령어 취소
  - **U** : 현재 라인을 변경 이전의 상태로 복구
- 대문자와 소문자 서로 변환 - **~(tilde)**
- 손실된 파일 복구 - **vi -r** 파일명

마지막 라인 모드에서 파일 관리 명령

새로운 파일에 현재 텍스트 저장

명령어	설 명
:*n*w 파일명	*n*라인의 텍스트를 파일에 저장
:*n*,*m*w 파일명	*n*에서 *m*라인의 텍스트를 파일에 저장

- 라인 번호 확인하는 방법
  - :→ 현재 커서set number → 전체 라인 번호 확인
  - :.= 의 라인 번호 확인

새로운 파일 읽어서 삽입

명령어	설 명
:r 파일명	커서가 위치한 아래 라인부터 파일 삽입
:*n*r 파일명	*n*라인 아래에 파일 삽입

새로운 파일 불러오기

명령어	설 명
:e 파일명	현재 작업중인 파일 내용을 저장하고 새로운 파일 불러옴
:e! 파일명	현재 작업중인 파일 내용을 저장하지 않고 새로운 파일 불러옴

마지막 라인 모드에서 검색 및 치환

마지막 라인 모드에서 검색

명령어	설 명
:f[문자]	동일 라인에서 검색
:/[단어]	앞 방향으로 단어 검색
:?[단어]	뒤 방향으로 단어 검색

명령어	설 명
소문자 n	동일한 검색 단어를 뒤 방향으로 반복하여 검색
대문자 N	동일한 검색 단어를 앞 방향으로 반복하여 검색

마지막 라인 모드에서 검색 (계속)

- 특수한 메타 문자를 사용하여 검색

명령어	설 명
:/^[문자열]	^ 검색 문자열을 라인의 시작으로 한정
:/[문자열]$	$ 검색 문자열을 라인의 마지막으로 한정
:/[문자].[문자]	. 어떤 한문자를 나타냄
:/[문자열][n-m]	[] 문자열의 집합
:/[문자열]*	* 0 이상의 모든 문자열 의미
:/\<[문자열]	\< 패턴을 단어의 시작으로 한정
:/\>[문자열]	\> 패턴을 단어의 마지막으로 한정

마지막 라인 모드에서 검색 (계속)

- 검색 동작 제어

명령어	설 명
`:ignorecase`	패턴 검색시 대소문자 구분하지 않음
`:wrapscan`	전체 파일에 대하여 패턴 검색

마지막 라인 모드에서 치환

```
:[범위 지정]s/old/new/[g][c]
```

- 범위 지정 옵션

옵 션	설 명
.(dot)	현재 라인을 의미
$	파일의 마지막 라인의 의미
%(1,$)	파일의 전체를 의미
.,$	현재 커서 위치에서 파일의 마지막 라인까지를 의미
n	*n*라인을 의미
+*n*	현재 라인을 시작으로 *n*번째 라인까지의 범위를 의미
-*n*	현재 라인을 시작으로 -*n*번째 라인까지의 범위를 의미
n,*m*	*n*라인에서 *m*라인까지의 범위를 의미

마지막 라인 모드에서 치환 (계속)

옵 션	설 명
g(Global)	범위 내의 모든 일치되는 패턴들을 찾아서 변경
c(Conforming)	여러 개의 동일한 패턴이 있는 경우 변경 유무 확인

vi의 편리한 환경 설정 옵션

:set [옵션]

옵 션	약 자	디폴트	설 명
autoindent	ai	noai	자동 들여쓰기
ignorecase	ic	noic	검색, 치환시 대소문자 구별하지 않음
magic	magic	magic	검색에서 메타 문자 사용할 수 있게 함
number	nu	nonu	편집기 라인 번호 표시
redraw	redraw	noredraw	각 문자를 항상 자기 위치에 나타냄
autowrite	aw	noaw	파일 이동 및 쉘로 빠져나올 때 강제 저장
mesg	mesg	mesg	문서 편집시 메시지 화면에 출력
showmode	smd	nosmd	현재 편집 모드의 상태를 화면에 표시
tabstop	ts	ts=8	Tab키의 공백 수
terse	terse	noterse	오류 메시지 간략 표현
window	wi	wi=23	vi 화면의 라인 수
wrapmargin	wm	wm=0	여백 크기 지정