



## 부록 vi 에디터 사용법

## ● vi 시작하기

## ▶ 기본 문법




- vi [ file ...]

## ● 입력 모드

[표 C-1] 입력 모드 전환 명령키

명령키	설명
i	커서의 위치에서부터 내용 입력하기
I	현재 행의 제일 앞에서부터 내용 입력하기
a	커서의 뒤에서부터 내용 입력하기
A	현재 행의 제일 마지막에서부터 내용 입력하기
o	커서 아래에 새 행을 추가하고 입력하기
O	커서 위에 새 행을 추가하고 입력하기

[표 C-2] 입력 모드에서 사용하는 명령키

명령키	설명
	다음 행으로 이동
	입력된 마지막 문자 지우기
	명령 모드로 전환

## Appendix vi 에디터 사용법

### ● 명령 모드

#### ▶ 커서 이동

[표 C-3] 커서 이동 명령키

명령키	설명	명령키	설명
h	커서를 왼쪽으로 이동	\$	행의 끝
j	커서를 아래로 이동	H	화면의 맨 위 행
k	커서를 위로 이동	L	화면의 맨 아래 행
l	커서를 오른쪽으로 이동	1g	문서의 제일 처음
b	왼쪽 단어의 앞	 + f	다음 페이지
w	오른쪽 단어의 앞	 + b	이전 페이지
e	단어의 끝	 + e	1행 스크롤 내리기
0 또는 ^	행의 처음	 + y	1행 스크롤 올리기

## Appendix      vi 에디터 사용법

### ▶ 삭제와 수정


[표 C-4] 삭제/수정 명령키

명령키	설명
x	커서 위치 문자 하나를 지운다.
dd	한 행 전체를 지운다(숫자와의 조합도 가능).
dw	한 단어를 지운다.
D	현재 행의 끝까지 지운다.
R	<b>Esc</b> 키를 눌러 명령 모드로 전환할 때까지 수정한다.
r	한 글자만 수정한 후 자동으로 명령 모드로 복귀한다.
s	커서 위치의 한 글자를 삭제하고 입력 모드로 복귀한다.
cw	한 단어의 수정, 명령을 입력하면 한 단어가 지워지고 입력 모드로 변경한다.
cc	한 행의 수정, 한 행이 지워지고 입력 모드로 변경한다.

## Appendix      vi 에디터 사용법

### ▶ 블록 설정과 문자열 복사

[표 C-5] 블록 설정과 문자열 복사 명령키

명령키	설명
v	블록을 만든다.
 +v	사각형 모양으로 블록을 만든다.
y	블록으로 지정한 문자열들을 버퍼에 저장한다.
yy	한 행을 버퍼에 저장한다.
p	커서의 뒤에 버퍼의 내용을 붙여넣는다.
P	커서의 앞에 버퍼의 내용을 붙여넣는다.

## Appendix      vi 에디터 사용법

### ▶ 파일 저장과 끝내기

[표 C-6] 파일 저장 및 종료 시 명령

명령	설명
:w	현재 문서를 저장한다.
:w 파일명	현재 문서를 지정한 파일명으로 저장한다.
:q	vi를 종료한다. 이때 문서의 변경이 있었다면 수행되지 않는다.
:q!	문서의 변경된 사항을 무시하고 vi를 종료한다.
:wq	문서를 저장한 뒤 종료한다.

### ▶ 파일 불러오기

[표 C-7] 파일 불러오기 명령

명령	설명
:e 파일명	현재 파일 대신 지정한 이름의 파일을 편집한다.
:r 파일명	현재 위치에서 지정한 이름의 파일 내용을 읽어온다.
:n	초기 명령 행에 주어진 파일 중 다음 순서에 해당하는 파일을 편집한다.

## Appendix      vi 에디터 사용법

### ▶ 문자열 검색과 치환

[표 C-8] 문자열 검색 명령키

명령키	설명
n	아랫 방향으로 문자열을 검색한다.
N	윗 방향으로 문자열을 검색한다.

### ▶ 기타 명령들

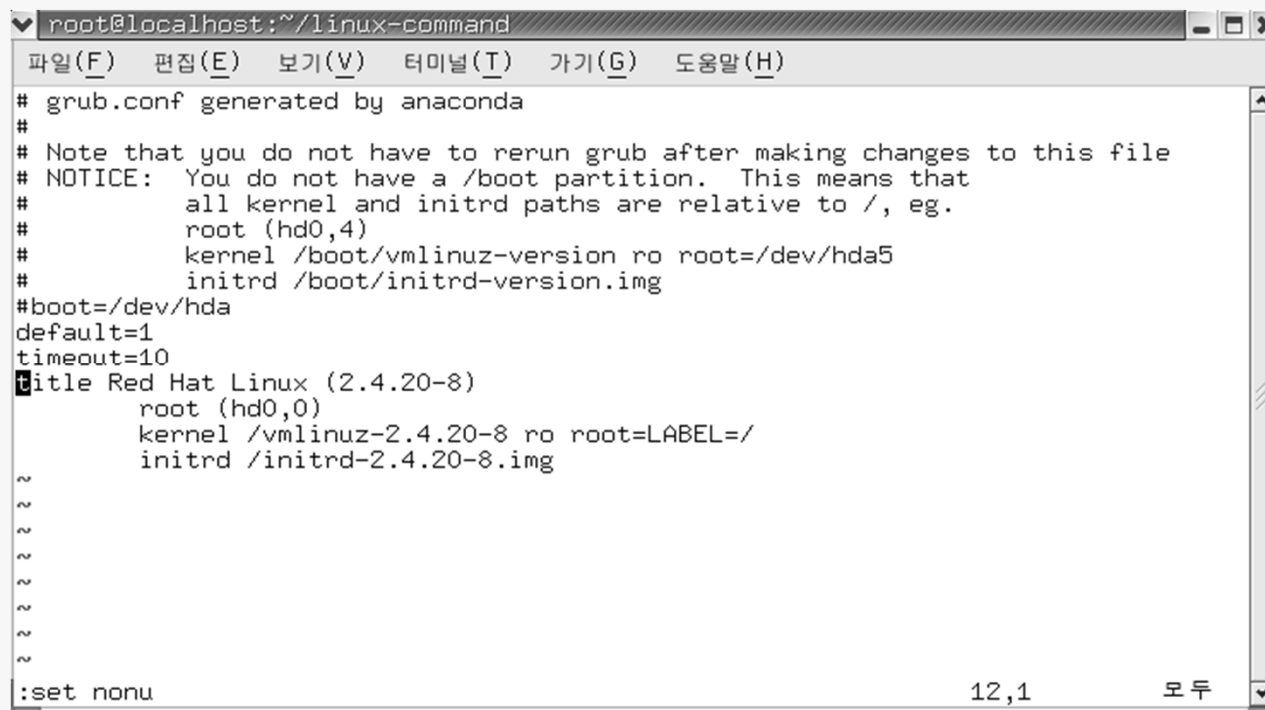
[표 C-9] 기타 명령

명령키	설명
<b>Ctrl</b> +L	화면을 다시 그린다. 이 명령은 화면에 메시지로 인해 화면이 어수선한 경우 또는 vi 사용 중에 연결된 모뎀과 간섭 현상이 있는 경우, 화면을 깨끗이 하는 데 유용하다.
!:<명령어>	vi를 중단하고 지정한 셸 명령을 수행한다.
!:<명령>	서브 셸에서 명령을 실행하고 현재 행 다음에 결과를 삽입한다.

## Appendix vi 에디터 사용법

### ▶ 환경 설정

- 기본 환경 설정 내용이 /usr/share/vim/vim61/vimrc\_example.vim에 기록되어 있다.
- /etc/grub/grub.conf 파일을 열어보면 다음과 같이 나타난다.



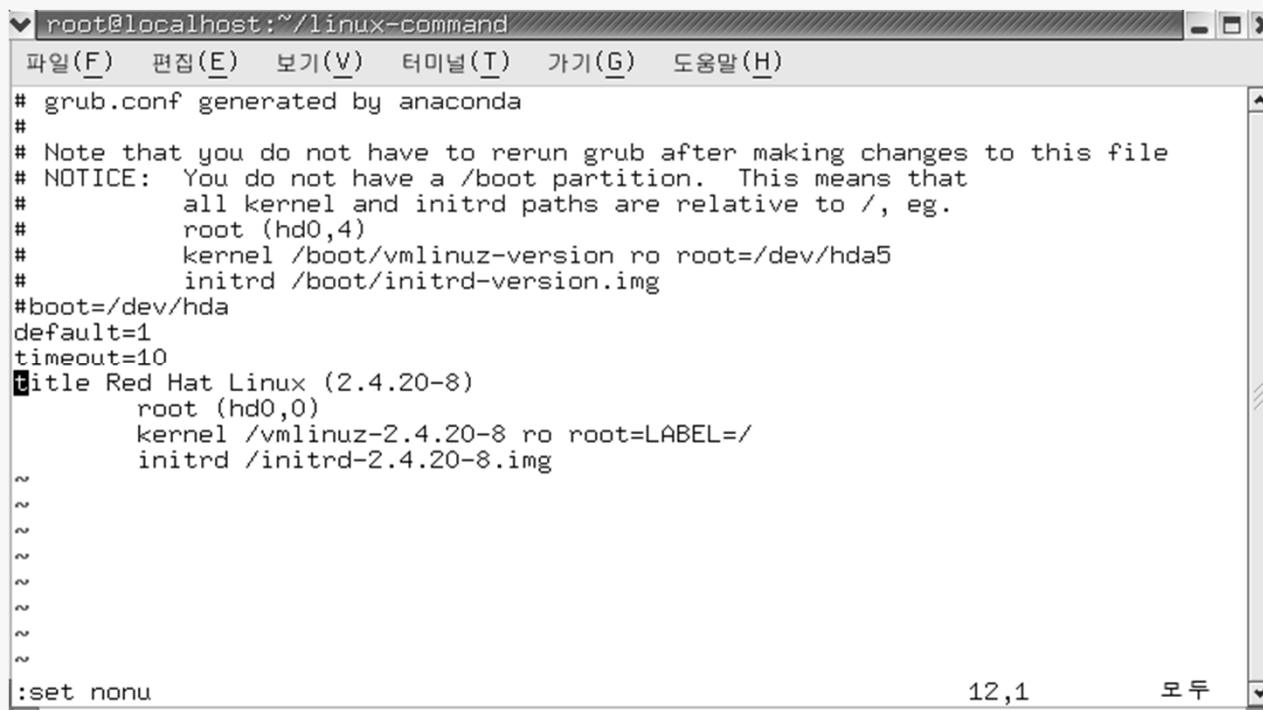
```
root@localhost:~/linux-command
파일(F) 편집(E) 보기(V) 터미널(T) 가기(G) 도움말(H)
# grub.conf generated by anaconda
#
# Note that you do not have to rerun grub after making changes to this file
# NOTICE:  You do not have a /boot partition.  This means that
#           all kernel and initrd paths are relative to /, eg.
#           root (hd0,4)
#           kernel /boot/vmlinuz-version ro root=/dev/hda5
#           initrd /boot/initrd-version.img
#boot=/dev/hda
default=1
timeout=10
title Red Hat Linux (2.4.20-8)
    root (hd0,0)
    kernel /vmlinuz-2.4.20-8 ro root=LABEL=/
    initrd /initrd-2.4.20-8.img
~
~
~
~
~
~
~
~
:set nonu
12,1  모두
```



## Appendix vi 에디터 사용법

### ▶ 환경 설정

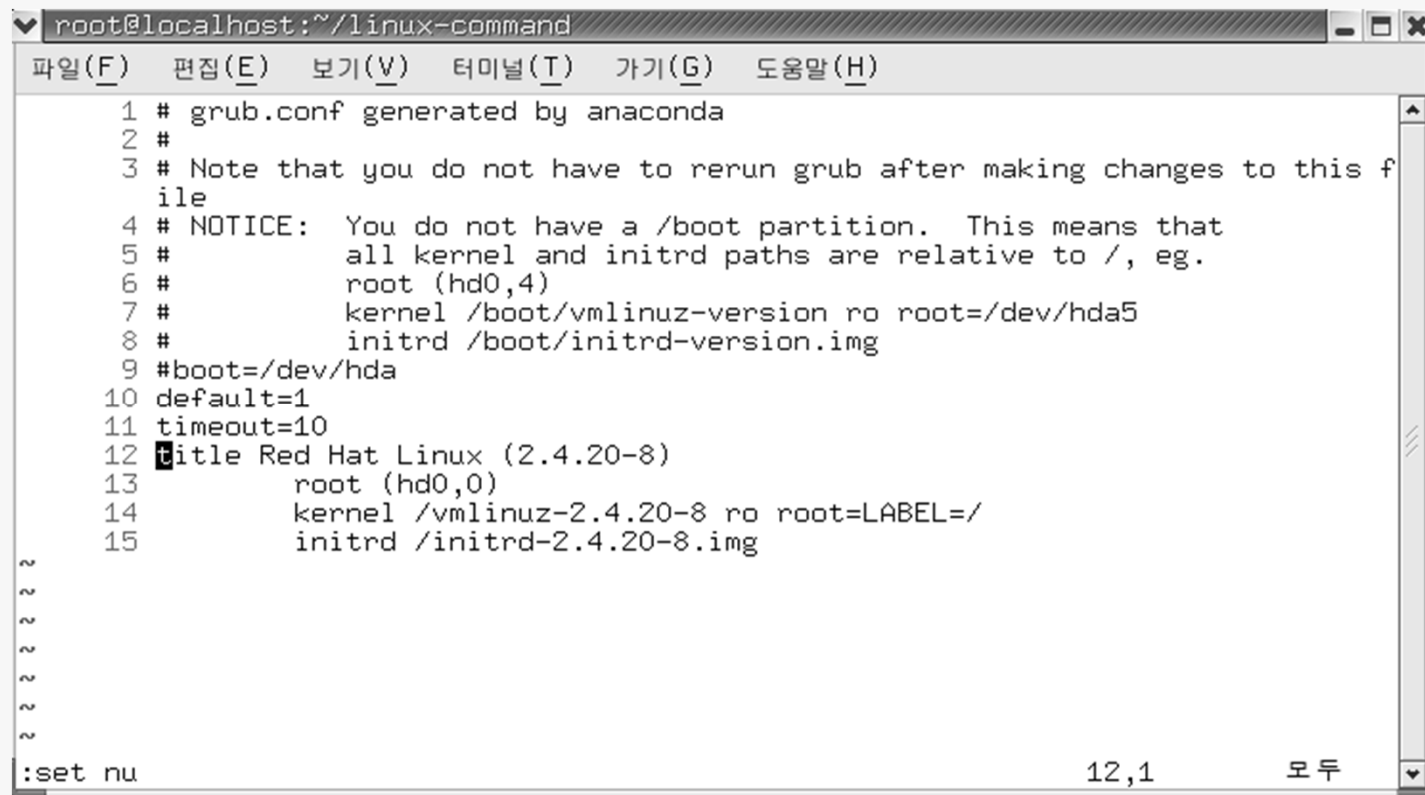
- 기본 환경 설정 내용이 /usr/share/vim/vim61/vimrc\_example.vim에 기록되어 있다.
- /etc/grub/grub.conf 파일을 열어보면 다음과 같이 나타난다.



```
root@localhost:~/linux-command
파일(F) 편집(E) 보기(V) 터미널(T) 가기(G) 도움말(H)
# grub.conf generated by anaconda
#
# Note that you do not have to rerun grub after making changes to this file
# NOTICE:  You do not have a /boot partition.  This means that
#           all kernel and initrd paths are relative to /, eg.
#           root (hd0,4)
#           kernel /boot/vmlinuz-version ro root=/dev/hda5
#           initrd /boot/initrd-version.img
#boot=/dev/hda
default=1
timeout=10
title Red Hat Linux (2.4.20-8)
    root (hd0,0)
    kernel /vmlinuz-2.4.20-8 ro root=LABEL=/
    initrd /initrd-2.4.20-8.img
~
~
~
~
~
~
~
~
:set nonu
12,1  모두
```

## Appendix vi 에디터 사용법

- 행 번호 출력
  - ✓ set nu를 입력
  - ✓ 번호를 없애는 명령은 set nonu다.



```
root@localhost:~/linux-command
파일(F) 편집(E) 보기(V) 터미널(T) 가기(G) 도움말(H)
1 # grub.conf generated by anaconda
2 #
3 # Note that you do not have to rerun grub after making changes to this f
4 # NOTICE: You do not have a /boot partition. This means that
5 #         all kernel and initrd paths are relative to /, eg.
6 #         root (hd0,4)
7 #         kernel /boot/vmlinuz-version ro root=/dev/hda5
8 #         initrd /boot/initrd-version.img
9 #boot=/dev/hda
10 default=1
11 timeout=10
12 title Red Hat Linux (2.4.20-8)
13     root (hd0,0)
14     kernel /vmlinuz-2.4.20-8 ro root=LABEL=/
15     initrd /initrd-2.4.20-8.img
:~
:~
:~
:~
:~
:~
:~
:set nu
12,1 모두
```

# Appendix      vi 에디터 사용법

## ▪ 주요 환경 설정 명령

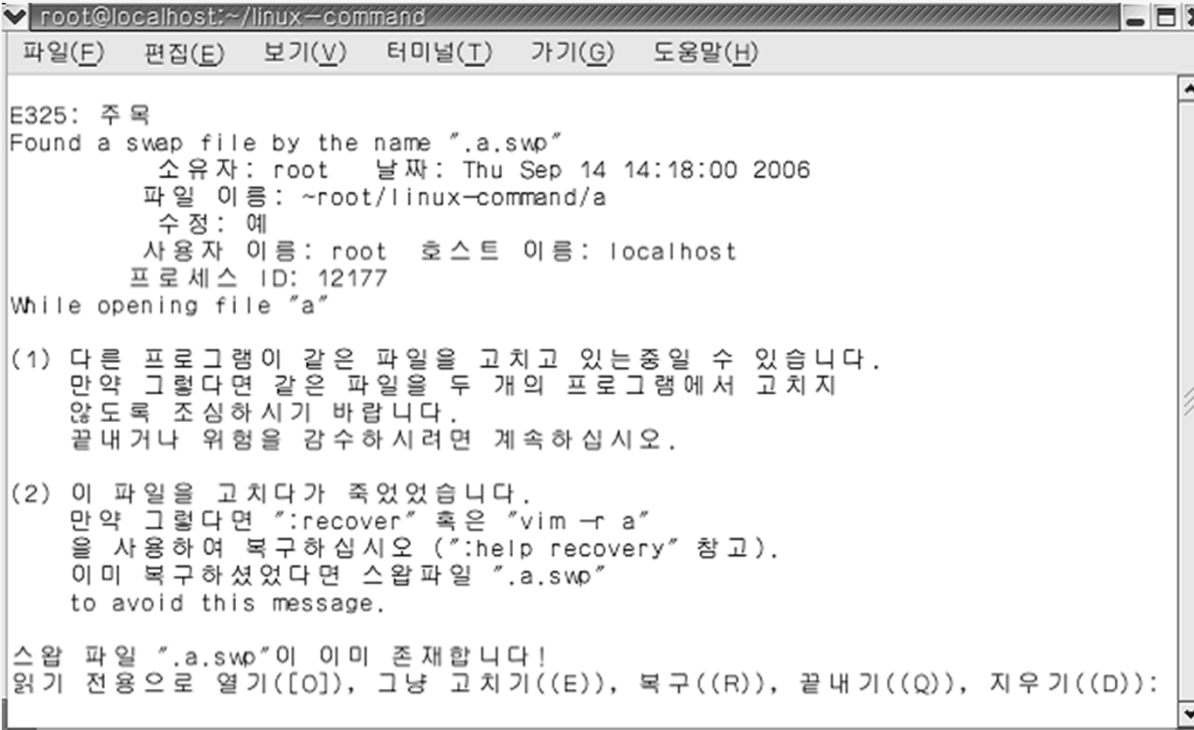
[표 C-10] 환경 설정 명령

명령	설명
set autoindent	자동으로 들여쓰기를 한다.
set cindent	C 언어 프로그램 작성 시 자동으로 들여쓰기를 한다.
set autowrite	파일 수정 시 자동으로 저장한다.
set smartindent	좀더 똑똑한 들여쓰기를 위한 옵션이다.
set textwidth=79	만약 79번째 글자를 넘어가면 다음 행으로 넘어간다.
set wrap	자동으로 <CR>을 삽입하여 다음 행으로 넘어간다.
set nowrapscan	검색할 때 문서의 끝에서 다시 처음으로 돌아가지 않는다.
set nobackup	백업 파일을 만들지 않는다.
set visualbell	키를 잘못 눌렀을 때 뽁 소리를 내는 대신 번쩍이게 한다.
set ruler	화면 우측 하단에 현재 커서의 위치(행, 칸)를 보여준다.
set tabstop=4	탭을 눌렀을 때 8칸 대신 4칸 이동하도록 한다.
set shiftwidth=4	자동 들여쓰기를 할 때 4칸 들여쓰도록 한다.
syntax on	컬러 터미널인 경우 컬러로 표시된다(↔ syntax off)
set ignorecase	문자열을 검색할 때 대소문자를 구별한다.
set showmode	상태 행에 현재 편집 모드를 출력한다.

## Appendix vi 에디터 사용법

### ▶ vi 스왑 파일

- vi에서 파일을 편집하면 저장하기 전까지의 모든 내용은 메모리에 임시로 저장된다.
- 파일로 저장하지 않고 터미널을 종료하는 경우에는 자동으로 스왑 파일을 생성하여 준다.



```
root@localhost:~/linux-command
파일(F) 편집(E) 보기(V) 터미널(T) 가기(G) 도움말(H)

E325: 주목
Found a swap file by the name ".a.swp"
    소유자: root   날짜: Thu Sep 14 14:18:00 2006
    파일 이름: ~root/linux-command/a
    수정: 예
    사용자 이름: root   호스트 이름: localhost
    프로세스 ID: 12177
While opening file "a"

(1) 다른 프로그램이 같은 파일을 고치고 있는중일 수 있습니다.
    만약 그렇다면 같은 파일을 두 개의 프로그램에서 고치지
    않도록 조심하시기 바랍니다.
    끝내거나 위험을 감수하시려면 계속하십시오.

(2) 이 파일을 고치다가 죽었습니다.
    만약 그렇다면 ":recover" 혹은 "vim -r a"
    을 사용하여 복구하십시오 (":help recovery" 참고).
    이미 복구하셨었다면 스왑파일 ".a.swp"
    to avoid this message.

스왑 파일 ".a.swp"이 이미 존재합니다!
읽기 전용으로 열기([O]), 그냥 고치기([E]), 복구([R]), 끝내기([Q]), 지우기([D]):
```

**Thank you**

