
3장. 파일편집

문서편집기(Text Editor)

- **Text** 형식의 문서를 새로 만들고 수정하고 하는 일련의 작업을 하는데 쓰는 유틸리티 프로그램
- 텍스트 에디터의 종류
 - ◆ 대화형 유틸리티
 - 라인 에디터(Line Editor)
 - "ed", "ex"
 - 스크린(Screen Editor)
 - "vi", "emacs", "pico"
 - ◆ 비대화형 에디터
 - 스트림 에디터 - "sed"
- 모든 에디터는 현재 수정한 값들이 원본 파일에 그대로 저장되지 않고 임시공간인 버퍼(buffer)에 저장

라인 에디터 ed 와 ex

■ 사용 형식

- ◆ 명령어 다음에 파일명을 입력

- **ed [filename] 또는 ex [filename]**

■ 2 가지 모드 사용

- ◆ 명령모드 - 모든 입력이 명령어로 간주

- ◆ 입력모드 - 텍스트 입력

- ◆ 입력 모드 → 명령 모드 - 인터럽트(Ctrl-C)

- ◆ 명령 모드 → 입력 모드 - “a”(append), “i”(insert), “c”(change)

■ 명령 프롬프트 - “?”(ed) 또는 “:”(ex)

■ 저장하고 종료하기

- ◆ 실제 원본파일 대신 버퍼에서만 작업하므로 명시적으로 저장 필요

- ◆ “w” 명령어 - 현재 버퍼에 있는 모든 내용을 저장

- ◆ “q”(quit) 명령어 - 편집 작업 종료

라인 에디터 ed 와 ex

■ 기본 명령어

명령	의미	명령	의미
a	문자열 추가(append)	c	지정 행의 문자열 치환(change)
i	지정 행에 문자열 삽입(insert)	d	지정 행 삭제(delete)
s	문자열 치환(substitute)	p	현재 내용 출력(print)
e	다른 파일 읽어들임(enter)	+	아래로 행 이동
r	다른 파일 내용 삽입	-	위로 행 이동
w	저장	/	아래로 문자열 검색
l	출력할 수 없는 문자 표시(list)	?string?	위로 문자열 검색
k	특정 행을 알파벳으로 지정	m	특정 행 이동(move)
=	지정 행 번호 출력	u	이전 명령 취소(undo)

sed 스트림 에디터

■ 스트림 편집기

- ◆ 파일이나 표준 입력으로부터 버퍼로 읽어 각 행마다 편집 명령을 실행한 후 그 결과를 다시 표준 출력으로 표시하는 에디터
- ◆ "sed" 에디터의 가장 큰 특징은 원본 파일을 변경하지 않음
- ◆ 반복 작업이나 아주 큰 파일을 처리할 때 매우 유용

■ 명령 양식

sed	
일반형식	sed [options] filename
주요옵션	-f : 스크립트 파일을 불러옴 -e : 여러 개의 스크립트를 하나의 명령 행에 지정 -n : 명령어 지정 행을 제외하고 표준 출력하지 않음

■ 기본명령어

명령	의미	명령	의미
s	문자열 치환	c	지정 행의 문자열 치환
a	맨 마지막에 문자열 추가	p	출력
i	지정 행에 문자열 삽입	w	저장

vi 편집기

■ 시작과 종료

◆ vi 시작 명령

vi 또는 vim	
일반형식	vi [filename ...]
주요옵션	filename – 파일명은 여러 개, 와일드카드(*,?) 가능

➤ 읽기만 할 때는 “vi” 대신 “view” 사용

◆ vi 실행화면 – 다음페이지 참조

- 화면이 지워지고 사용자 파일이 버퍼로 읽어짐
- 커서는 버퍼의 첫 번째 라인 첫 문자에 위치
- 화면 아래에 현재 파일명과 라인 수, 전체 문자 수 표시
- 표시는 현재 공간을 의미

vi 편집기

■ vi 시작화면



The screenshot shows a terminal window titled 'cprog2@saps5:-'. The vi editor's start screen is displayed, featuring the text: 'VIM - Vi Improved', 'version 6.1.87', 'by Bram Moolenaar et al.', and 'Vim is open source and freely distributable'. It also includes a charity message: 'Help poor children in Uganda!' and instructions on how to use the editor's help system. At the bottom, it shows the current cursor position '0,0-1' and the file name 'All'. The status line at the very bottom reads '[영어][완성][두벌식]'.

```
cprog2@saps5:-  
  
VIM - Vi Improved  
version 6.1.87  
by Bram Moolenaar et al.  
Vim is open source and freely distributable  
  
Help poor children in Uganda!  
type :help iccf<Enter> for information  
  
type :q<Enter> to exit  
type :help<Enter> or <F1> for on-line help  
type :help version6<Enter> for version info  
  
0,0-1 All  
[영어][완성][두벌식]
```

vi 편집기

■ vi 종료

◆ vi 종료 명령

형식	의미
:q!	파일 내용을 저장하지 않고 종료
:wq! 또는 ZZ	파일 내용을 저장하고 종료

vi 편집기

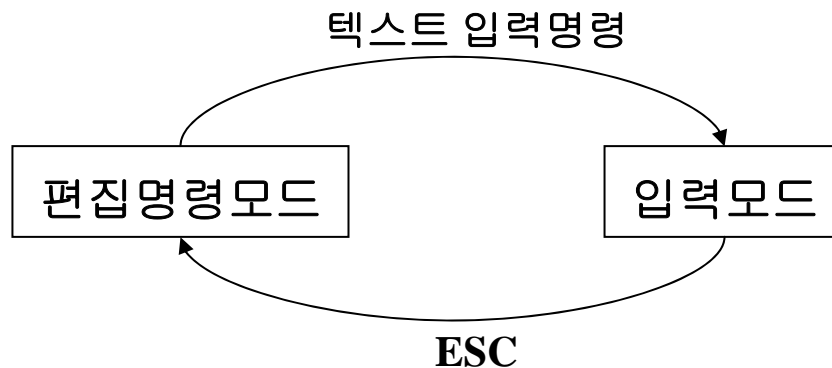
■ 입력모드와 편집명령모드

◆ 입력모드

- 파일의 내용을 입력할 수 있는 모드
- 입력모드전환 : **i, a, o**
- 사용자가 키보드를 치면 그 내용이 화면에 나타남

◆ 편집명령모드

- 편집 명령을 수행하는 모드
- 편집모드전환 : **ESC** 키
- 사용자의 키보드 입력이 화면에 나타나지 않고 명령 수행



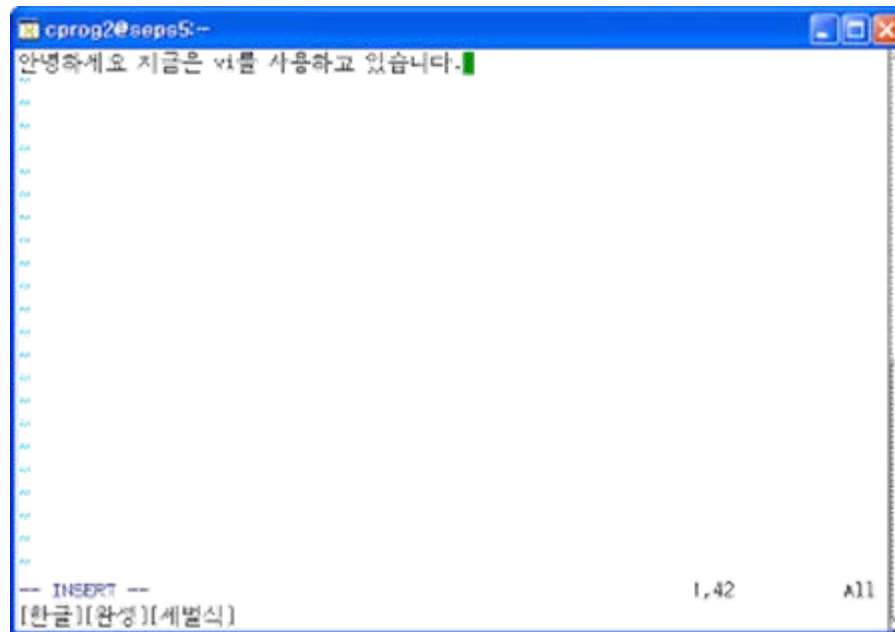
vi 편집기

■ 텍스트 입력

명 령	내 용
a	커서 바로 뒤에서 입력을 시작
A	커서가 위치한 행의 마지막부터 입력을 시작
i	커서가 위치한 곳부터 입력을 시작
I	커서가 위치한 행의 맨 앞에서 입력을 시작
o	커서가 위치한 행의 아래 행부터 입력을 시작
O	커서가 위치한 행의 위부터 입력을 시작

vi 편집기

■ 텍스트 입력 화면



vi 편집기

■ 커서의 이동

명 령	내 용
←, h	커서를 한 문자 왼쪽으로 이동
→, l	커서를 한 문자 오른쪽으로 이동
↑, k	커서를 한 행 위쪽으로 이동
↓, j	커서를 한 행 아래쪽으로 이동

vi 편집기

■ 커서의 이동

명 령	내 용
0	커서를 현재 행의 처음으로 이동
\$	커서를 현재 행의 마지막으로 이동
w	커서를 다음 단어의 처음으로 이동
b	커서를 이전 단어의 처음으로 이동
-	커서를 이전 행의 처음으로 이동
+	커서를 다음 행의 처음으로 이동
)	커서를 다음 문장의 처음으로 이동
(커서를 이전 문장의 처음으로 이동
}	커서를 다음 문단의 처음으로 이동
{	커서를 이전 문단의 처음으로 이동

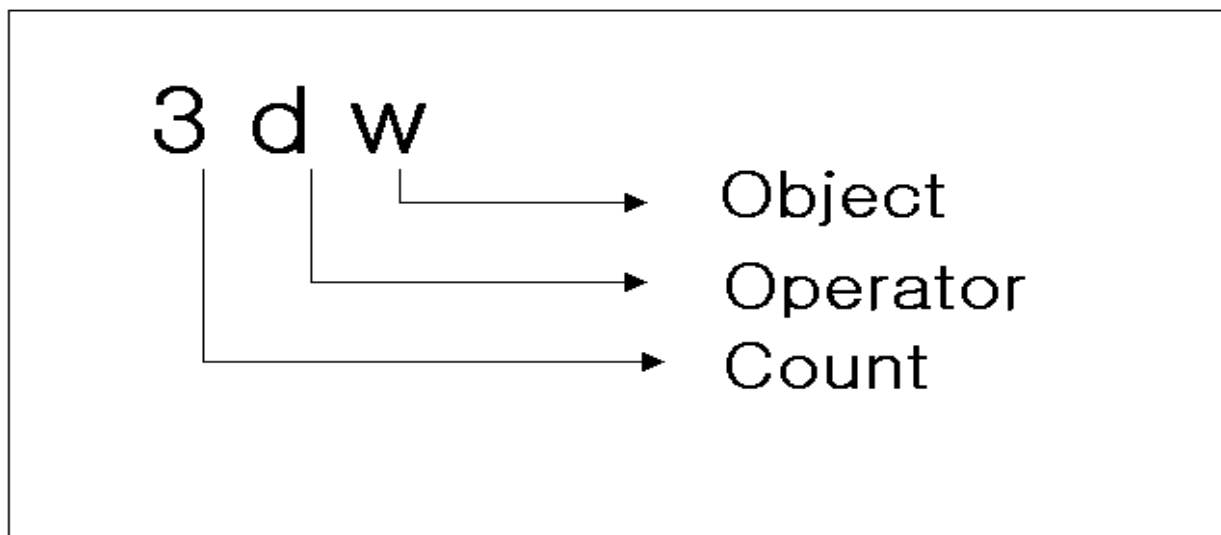
vi 편집기

■ 편집화면 이동

명 령	내 용
^f	한 화면 아래로 이동(Forward)
^b	한 화면 위로 이동(Backward)
^d	반 화면 아래로 이동(Down)
^u	반 화면 위로 이동(Up)
:0	파일의 처음으로 이동
:\$	파일의 끝으로 이동
:숫자	파일에서 지정한 숫자번째 행으로 이동

vi 편집기

■ 편집 명령어 구조



vi 편집기

■ 편집 내용 삭제

명 령	내 용
x	커서의 위치에 있는 문자 삭제
X	커서의 왼쪽에 있는 문자 삭제
dd	커서가 위치한 행 삭제
dw	한 단어 삭제
D	커서의 위치부터 오른쪽 내용 삭제
[숫자] 명령	숫자 수만큼 명령 수행

■ 사용 예

- ◆ **10dd** : 현재 커서 위치로부터 **10개의 행** 삭제
- ◆ **10x** : 현재 커서 위치로부터 **10개의 문자** 삭제
- ◆ **5dw** : 현재 커서 위치로부터 **5개의 단어** 삭제

vi 편집기

■ 편집내용 수정

명 령	내 용
r	한 문자 고쳐 쓰기
R	ESC 키를 칠 때까지 계속 고쳐 쓰기
cw	한 단어 고쳐 쓰기
cc	현재 행 고쳐 쓰기
C	커서의 위치부터 현재 행의 마지막까지 고쳐 쓰기
[숫자] 명령	숫자 수만큼 명령 수행

vi 편집기

■ 내용 잘라내기, 복사, 붙여넣기

명 령	내 용
y	커서가 위치한 행과 다음 행을 버퍼에 저장
Y	커서가 위치한 행을 버퍼에 저장
p	저장된 버퍼의 내용을 커서 다음 행에 붙여넣음
P	저장된 버퍼의 내용을 커서 이전 행에 붙여넣음
C	커서의 위치부터 현재 행의 마지막까지 고쳐 쓰기
[숫자] 명령	숫자 수만큼 명령 수행

■ 사용 예

◆ 5 행을 다른 행에 복사

➤ “5Y” 또는 “5yy” 한 후 이동할 행으로 커서를 움직인 후 “p” 수행

◆ 3 행을 다른 행으로 이동

➤ “3dd” 한 후 이동할 행으로 커서를 움직인 후 “p” 수행

vi 편집기

■ 그 밖의 편집 명령

명 령	내 용
u	수행한 명령을 취소. 수행하지 않은 상태로 되돌림(undo)
.	가장 최근에 수행한 명령을 다시 수행함
J	현재 행과 다음 행을 결합

vi 편집기

■ 문자열 검색 및 이동

명 령	내 용
/	정방향으로 문자열 검색
?	역방향으로 문자열 검색
n	다음 문자열 검색

◆ 문자열 검색에서 사용 가능한 기호들

기 호	내 용
*	모든 문자와 대응
.	한 문자와 대응
^	행의 첫 문자와 대응
\$	행의 마지막 문자와 대응
[]	대응되는 문자 범위 지정
[^]	범위에 해당하는 것을 제외한 것과 대응

vi 편집기

■ 정규 표현식(regular expression) 의 예

기 호	내 용
A*	A로 시작하는 모든 문자와 대응
A.	A로 시작하면서 다음에 한 문자가 오는 문자열과 대응
^A	행의 첫 문자가 A인 문자열과 대응
A\$	행의 마지막 문자가 A인 문자열과 대응
A[a-c]	A 다음에 a, b, 또는 c 가 나오는 문자열과 대응

vi 편집기

■ ex 명령어 사용

- ◆ ":"(콜론)을 사용
- ◆ 파일 전체나 행 단위로 파일을 제어할 때 사용
- ◆ 라인번호를 이용하여 여러 라인을 복사하거나 이동할 때 유용
- ◆ 정규 표현식 사용 가능
 - "."은 현재 줄, "\$"는 마지막 줄, "%"는 모든 줄 지정

■ 사용 예

- ◆ :s /원래문자열/바꿀문자열
 - 커서가 위치한 현재 줄에서 첫번째 해당하는 문자열을 바꿈
- ◆ :s /원래문자열/바꿀문자열/g
 - 커서가 위치한 현재 줄에서 해당하는 모든 문자열을 바꿈
- ◆ :1,10s /원래문자열/바꿀문자열/g
 - 첫째줄에서 열번째 줄에서 해당하는 모든 문자열을 바꿈
- ◆ : 10d
 - 현재 줄에서 10 줄을 삭제

vi 편집기

■ 기타 ex 명령어

◆ 편집기 내에서 일반 셸 명령어를 사용 가능 - "!"를 사용

➤ **:! ls -la** : 현재 디렉토리의 파일 목록 출력

◆ 외부에 있는 파일을 편집기 내로 불러올 때 "r" 명령어 사용

➤ **:r filename** : filename 파일의 내용을 현재 줄 다음부터 삽입

◆ 다른 파일로 이동

➤ **:e filename** : filename 파일로 이동

➤ **:e #** : 이전 파일로 복귀

vi 편집기

■ 사용 환경 설정

◆ “vi” 는 기본적인 환경 설정 값을 제공

◆ 설정 방법

➤ .exrc 또는 .vimrc 와 같은 파일을 작성하여 설정 값 지정

➤ 편집모드에서 “: set 옵션=값” 형식의 명령 사용

■ 대표적인 환경 설정 변수

옵 션	내 용	기본값
all	모든 현재 설정값을 표시	없음
autoindent	자동 들여쓰기 설정	autoindent
shiftwidth	자동 들여쓰기의 여백값	8
number	화면에 행 번호 표시	nonumber
tabstop	탭의 간격을 설정	8
showmode	삽입모드 표시 여부	showmode
wrapmargin	오른쪽 여백 지정	0

vi 편집기

□ 환경 설정 사용 예

◆ : set all

: 현재 설정되어 있는 값들을 모두 출력



```
cprog2@seps5:-
:set all
--- Options ---
aleph=224          nofknap          modelines=5        switchbuf=
noallourevis       foldclose=        modifiable        syntax=c
noaltkeymap        foldcolumn=0     modified          tabstop=8
autoindent         foldenable       more             tagsearch
noautoread         foldexpr=0       mouse            taglength=0
noautoresize       foldignore=#     mousenodel=extend tagrelative
noautoresizeall    foldlevel=0      mousetime=500    tagstack
background=dark    foldlevelstart=-1 number          term=xterm
backspace=2        foldmethod=manual nopaste          termencoding=
nobackup           foldnlines=1     pastetoggle=      naterse
backupcopy=auto    foldnestmax=20   patche>pr=        textauto
backupexts=""      formatprg=       patchmode=        notextmode
backupskip=/tmp/*  nogdefault       previewheight=12  textwidth=0
nabinary          helpheight=20    nopreviewwindow   thesaurus=
nabomb            nohidden         printdevice=       notildeop
bufhidden=         history=50       printfont=courier timeout
buflisted         nofknap          printoptions=     timeoutlen=1000
buftype=          nohiknap         noreadonly        notitle
cdpath=.,          hisearch         renop             titlelen=85
cedit="F"         noicon           report=2          titlestring=
charconvert=       iconstring=      norevis          nontimeout
-- More --
[영어] [완성] [두벌식]
```

◆ : set tabstop=4 : 탭 간격을 4 문자로(기본값은 8)

◆ : set number : 행 번호 출력

◆ : set nonumber : 행 번호 출력하지 않음

Pico 편집기

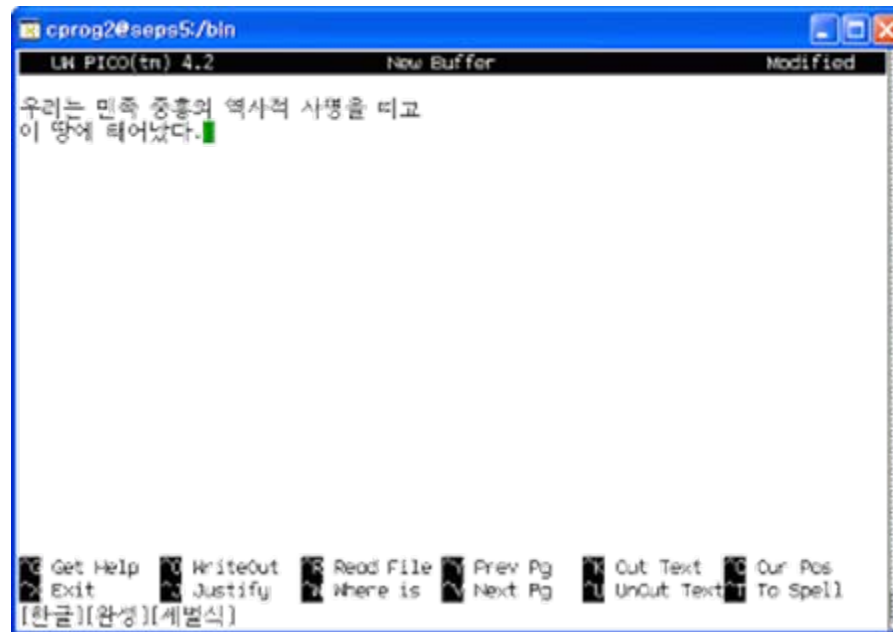
■ 초보자용 문서편집기

◆ vi와는 달리 Windows의 메모장과 비슷하게 쉽게 사용

◆ 사용 명령

➤ **\$ pico [파일이름]**

◆ 사용 화면



```
cprog2@seps5:/bin
LM PICO(tm) 4.2      New Buffer      Modified
우리는 민족 중흥의 역사적 사명을 띠고
이 땅에 태어났다.
[한글][완성][개별식]
```

Pico 편집기

■ pico 편집기의 단축키들

명 령	내 용
CTRL+g	도움말 보기
CTRL+o	파일 저장
CTRL+x	Pico 빠져나오기
CTRL+a	현재 행의 처음으로 이동
CTRL+e	현재 행의 처음으로 이동
CTRL+v	이전 페이지로 이동(page-up)
CTRL+y	다음 페이지로 이동(page-down)
CTRL+w	문자열 찾기
CTRL+d	현재 커서 위치의 한 글자 삭제
CTRL+k	한 줄 삭제
CTRL+u	마지막으로 삭제된 줄 복구

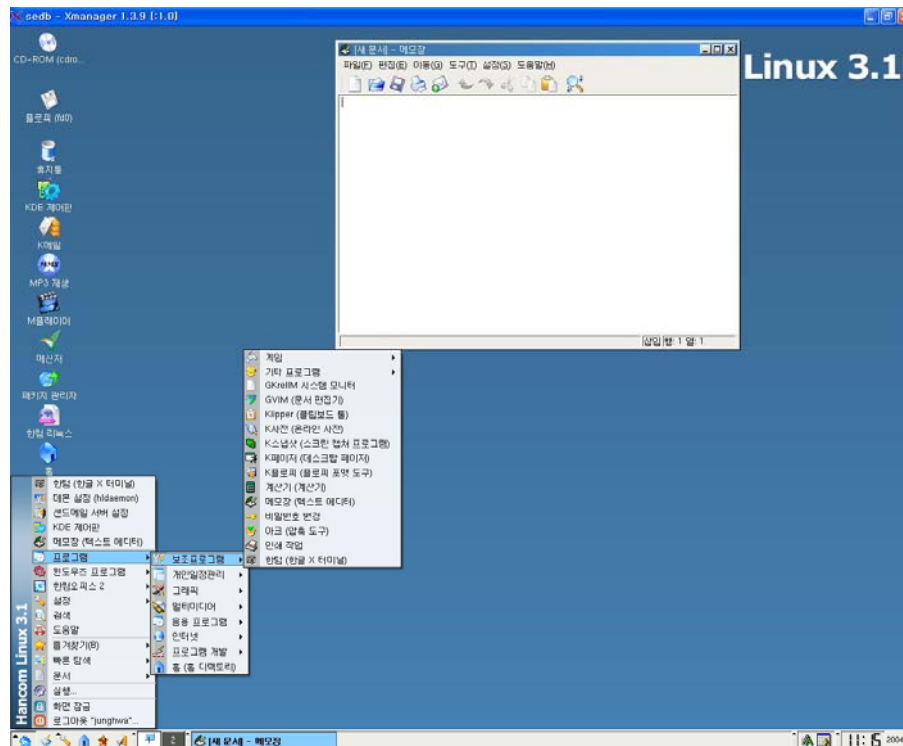
X window 환경에서의 문서편집기

■ 리눅스의 GUI 환경

◆ X Window : 하부 그래픽 윈도우 라이브러리

◆ KDE와 GNOME : 대표적인 GUI 환경

➤ 상위 툴킷/윈도우시스템에 따라 다름



KEDIT

■ 특징

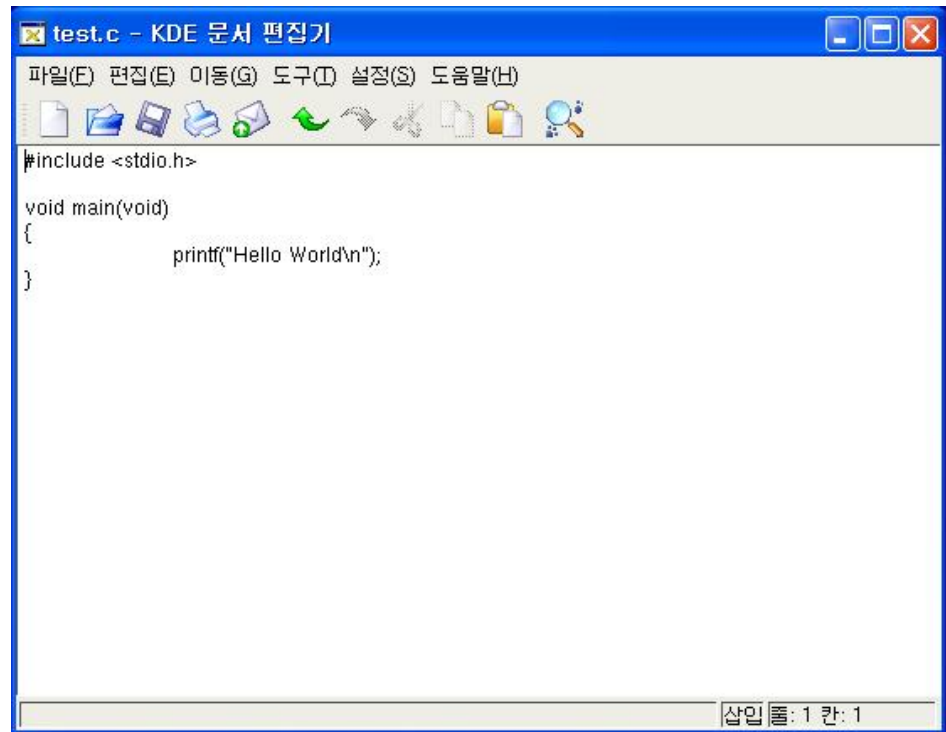
◆ GUI 기반 문서편집기

➤ KDE 및 Qt 기반

■ 사용법

\$ kedit [파일이름] &

■ KEDIT 실행 화면



■ kedit 의 기능

- ◆ 메뉴 선택 시 마우스와 단축키 사용 가능
- ◆ 빠른 도구 모음에는 풍선 도움말 제공

■ 주 메뉴

메뉴	내 용
File	저장하기, 불러오기 등의 파일관련 메뉴
Edit	잘라내기, 붙여넣기 등의 편집관련 메뉴
Go	라인이동 관련 메뉴
Tools	철자검사기와 같은 도구관련 메뉴
Settings	kedit 환경설정 메뉴
Help	도움말 관련 메뉴

GEDIT

■ 특징

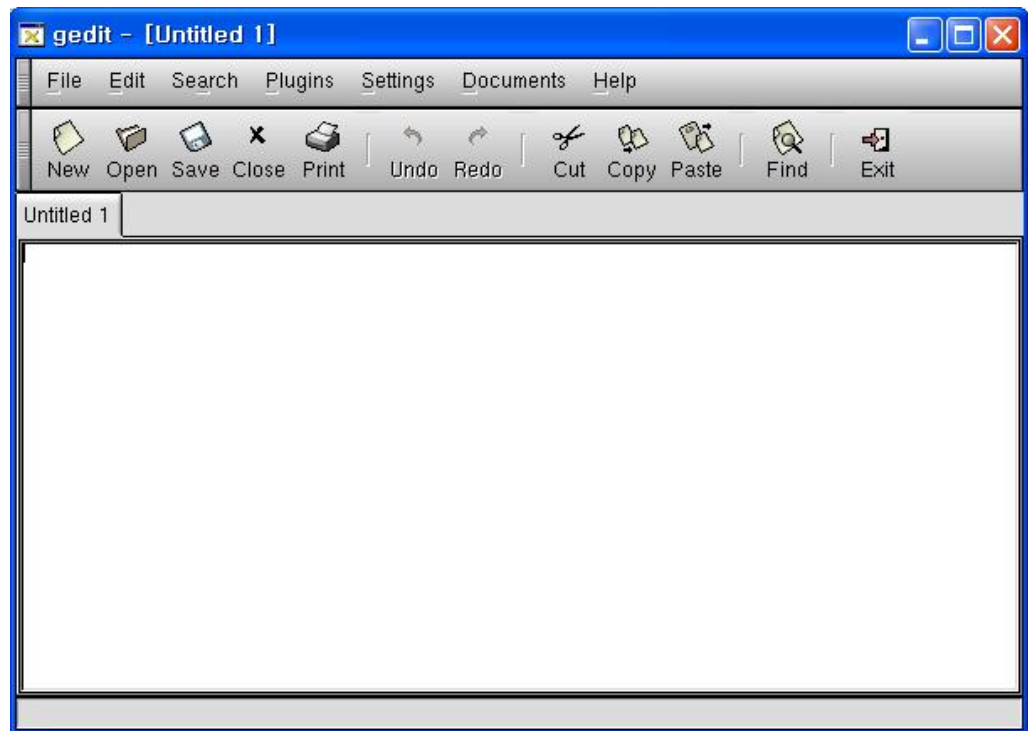
◆ GUI 기반 문서편집기

➤ GNOME 및 GTK+ 기반

■ 사용법

\$ gedit [파일이름] &

■ GEDIT 실행 화면



KATE

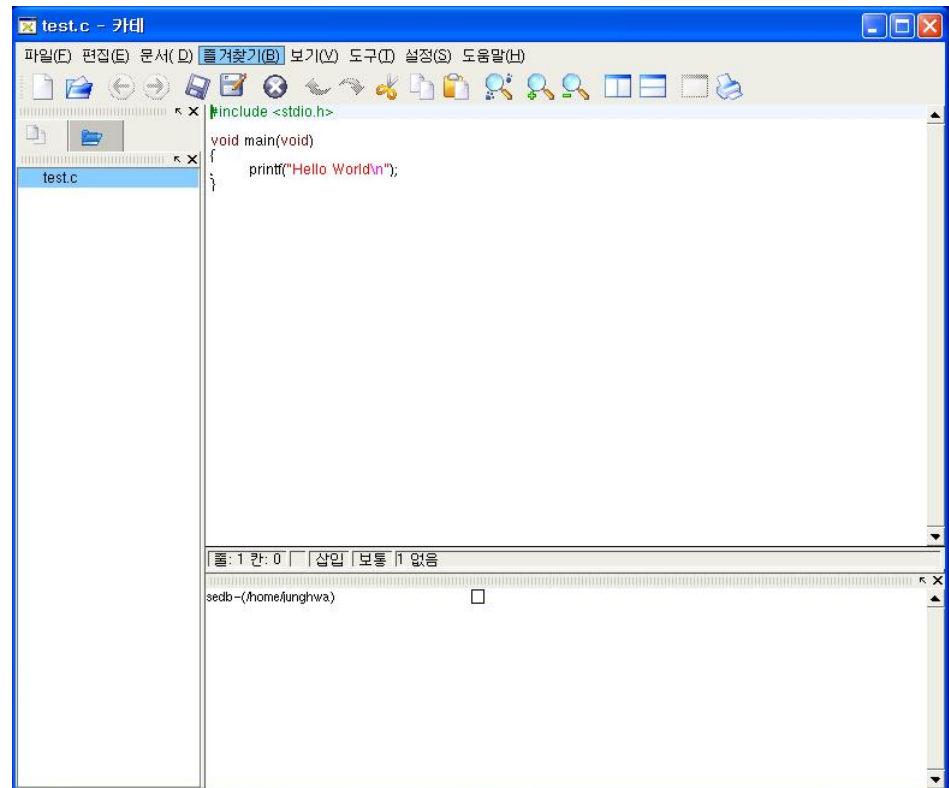
■ 특징

- ◆ GUI 기반 문서편집기
- ◆ 화면 분할 기능
- ◆ 셸 창을 별도로 제공

■ 사용법

`$ kate [파일이름] &`

■ KATE 실행 화면



창을 가로로 분할한 예

