



응용 SW 기초 활용 기술 part 1

# 리눅스 프로그래밍 개발도구



한국기술교육대학교  
온라인평생교육원



- 프로그래밍 코딩 도구
- vi 편집기와 gcc 컴파일러

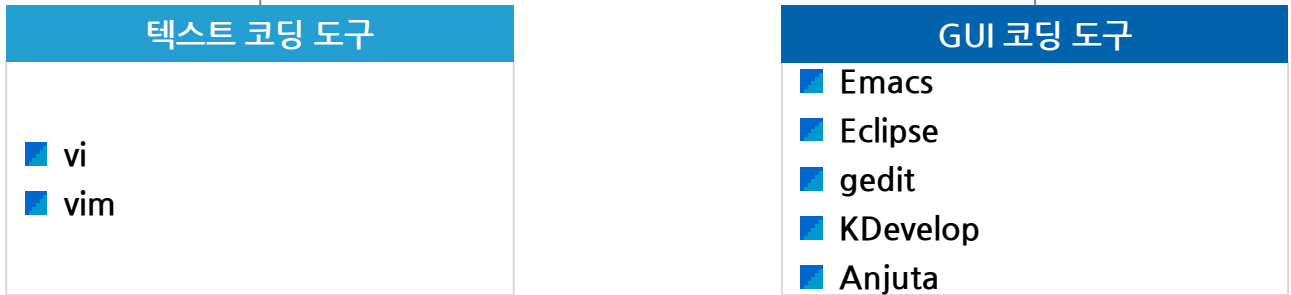


- 페도라 리눅스에서 사용할 수 있는 다양한 프로그래밍 코딩 도구에 대하여 설명할 수 있다.
- 페도라 리눅스의 vi 편집기 및 gcc 컴파일러를 이용하여 프로그래밍을 작성할 수 있다.

# 프로그래밍 코딩 도구

## 1 텍스트 프로그래밍 코딩 도구

### 대표적인 프로그래밍 코딩 도구



+ 페도라 리눅스에서 프로그래밍을 작성할 때는 **텍스트 기반의 프로그래밍 도구**를 사용하는 것을 추천함

- 프로그래밍 작성에 있어 **효율성**과 **속도**가 GUI 프로그래밍 도구보다 좋으며, 훨씬 **편리**하기 때문

vi(Visual editor)

- 1976년 빌조이 개발
- 키보드 입력을 통해 모든 명령 처리
- 리눅스 계열 운영체제 기본 탑재



## 프로그래밍 코딩 도구

### 2 GUI 프로그래밍 코딩 도구

#### 1) Emacs(Editong MACros)

Emacs  
(Editong MACros)

- 1976년 리처드 스톨만 제작
- 키보드로 모든 명령 처리
- 방대한 명령어(vim의 5배)
- 파일관리, 디버거, 웹브라우저, mail, ftp, news 기능 포함

장점

확장성이 좋음

단점

속도가 느림



Emacs는 기본적으로 설치되지 않으므로, 별도로 설치해야만 실행할 수 있음

➡ 패키지 설치 : `yum install emacs.x86_64`

```
*GNU Emacs* - emacs@korea.tech.com
File Edit Options Buffers Tools Help
Welcome to GNU Emacs, one component of the GNU/Linux operating system.
To follow a link, click Mouse-1 on it, or move to it and type RET.
To quit a partially entered command, type Control-g.
Important Help menu items:
Emacs Tutorial      Learn basic Emacs keystroke commands
Read the Emacs Manual View the Emacs manual using Info
(Non)Warranty       GNU Emacs comes with ABSOLUTELY NO WARRANTY
Copying Conditions   Conditions for redistributing and changing Emacs
More Manuals / Ordering Manuals How to order printed manuals from the FSF
Useful tasks:
Visit New File       Specify a new file's name, to edit the file
Open Home Directory Open your home directory, to operate on its files
Customize Startup     Change initialization settings including this screen
GNU Emacs 24.5.1 (x86_64-redhat-linux-gnu, GTK+ Version 3.16.2)
of 2015-04-22 on buildhw-10.phx2.fedoraproject.org
U:%%- *GNU Emacs* Top L1 (Fundamental)
For information about GNU Emacs and the GNU system, type C-h C-a.
```

Emacs 실행화면

## 프로그래밍 코딩 도구

### 2) GUI 프로그래밍 코딩 도구

#### 2) Eclipse

Eclipse

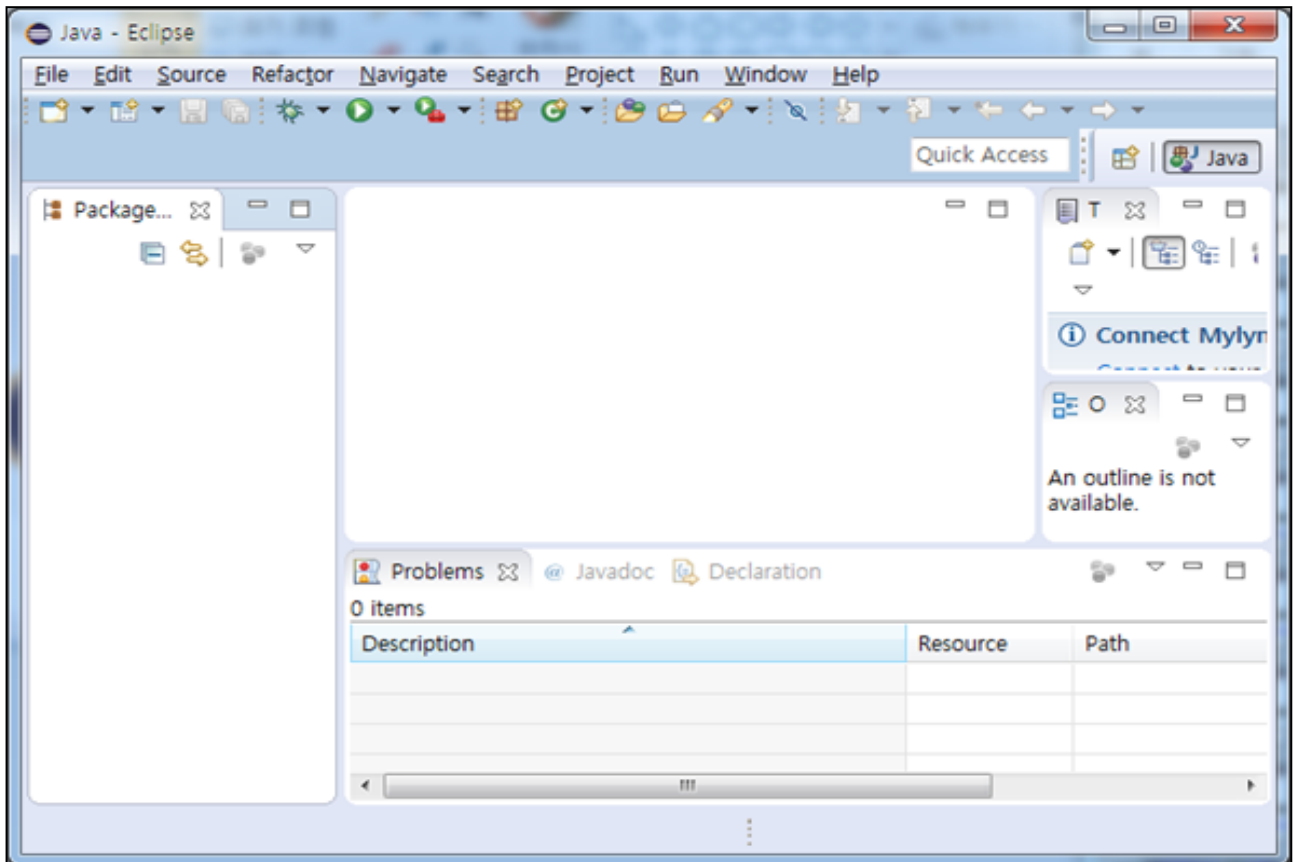
- Java 언어를 위한 IDE(Integrated Development Environment)
- 플러그인 확장을 통해 C/C++, COBOL 등의 다양한 언어를 사용할 수 있음

#### 장점

윈도우용 Eclipse와 사용법이 동일하여 편리함

#### 단점

리눅스 계열 운영체제에서는 속도가 느림



Eclipse 실행화면

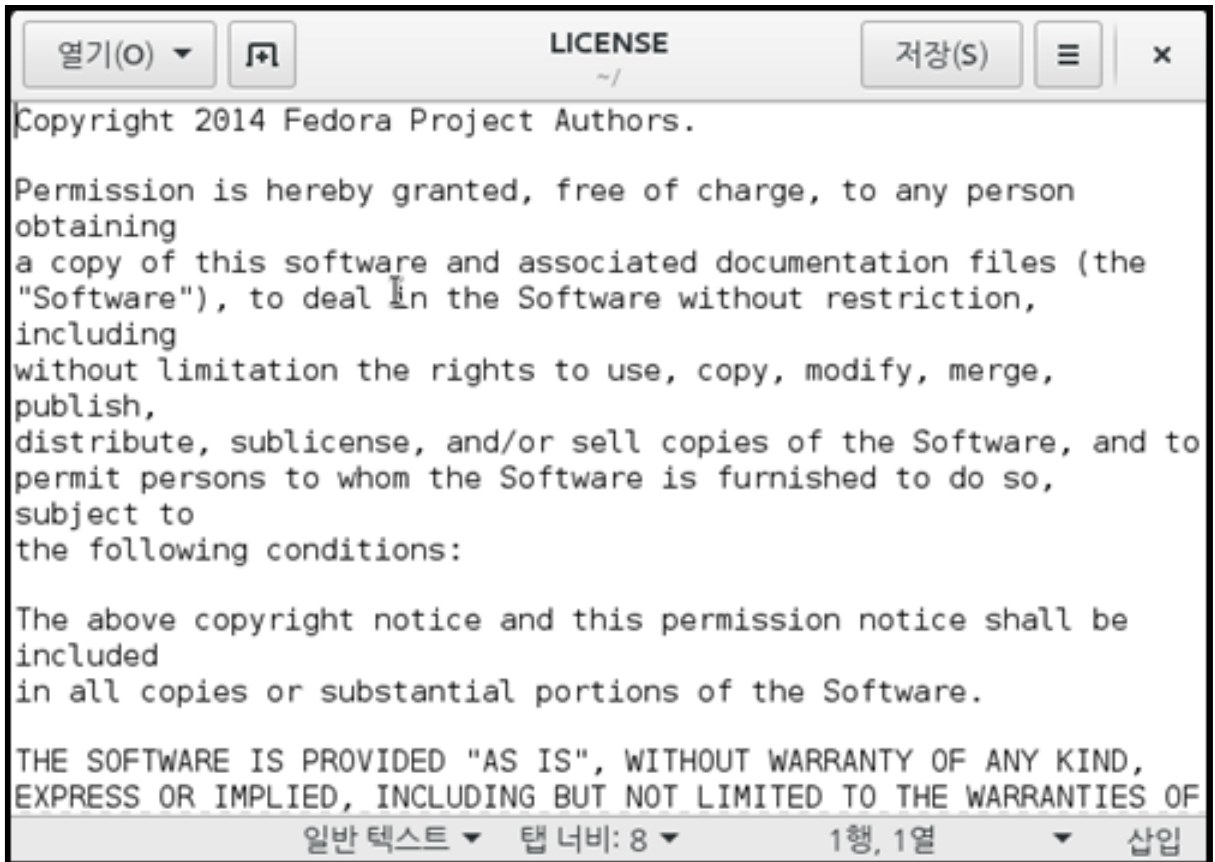
## 프로그래밍 코딩 도구

### 2 GUI 프로그래밍 코딩 도구

#### 3) gedit

gedit

- 가장 간단한 편집도구
- GNOME 데스크톱 환경에서 기본 제공
- 윈도우용 Editplus 와 거의 동일하여 윈도우에 익숙한 사용자들에게 편리함



The screenshot shows the gedit text editor window. The title bar includes buttons for '열기(O)' (Open), a file icon, 'LICENSE ~/', '저장(S)' (Save), a menu icon, and a close button 'x'. The main text area contains the following text:

```
Copyright 2014 Fedora Project Authors.  
  
Permission is hereby granted, free of charge, to any person  
obtaining  
a copy of this software and associated documentation files (the  
"Software"), to deal in the Software without restriction,  
including  
without limitation the rights to use, copy, modify, merge,  
publish,  
distribute, sublicense, and/or sell copies of the Software, and to  
permit persons to whom the Software is furnished to do so,  
subject to  
the following conditions:  
  
The above copyright notice and this permission notice shall be  
included  
in all copies or substantial portions of the Software.  
  
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,  
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
```

The status bar at the bottom shows '일반 텍스트' (Plain Text), '탭 너비: 8' (Tab width: 8), '1행, 1열' (1 line, 1 column), and '삽입' (Insert).

gedit 실행화면

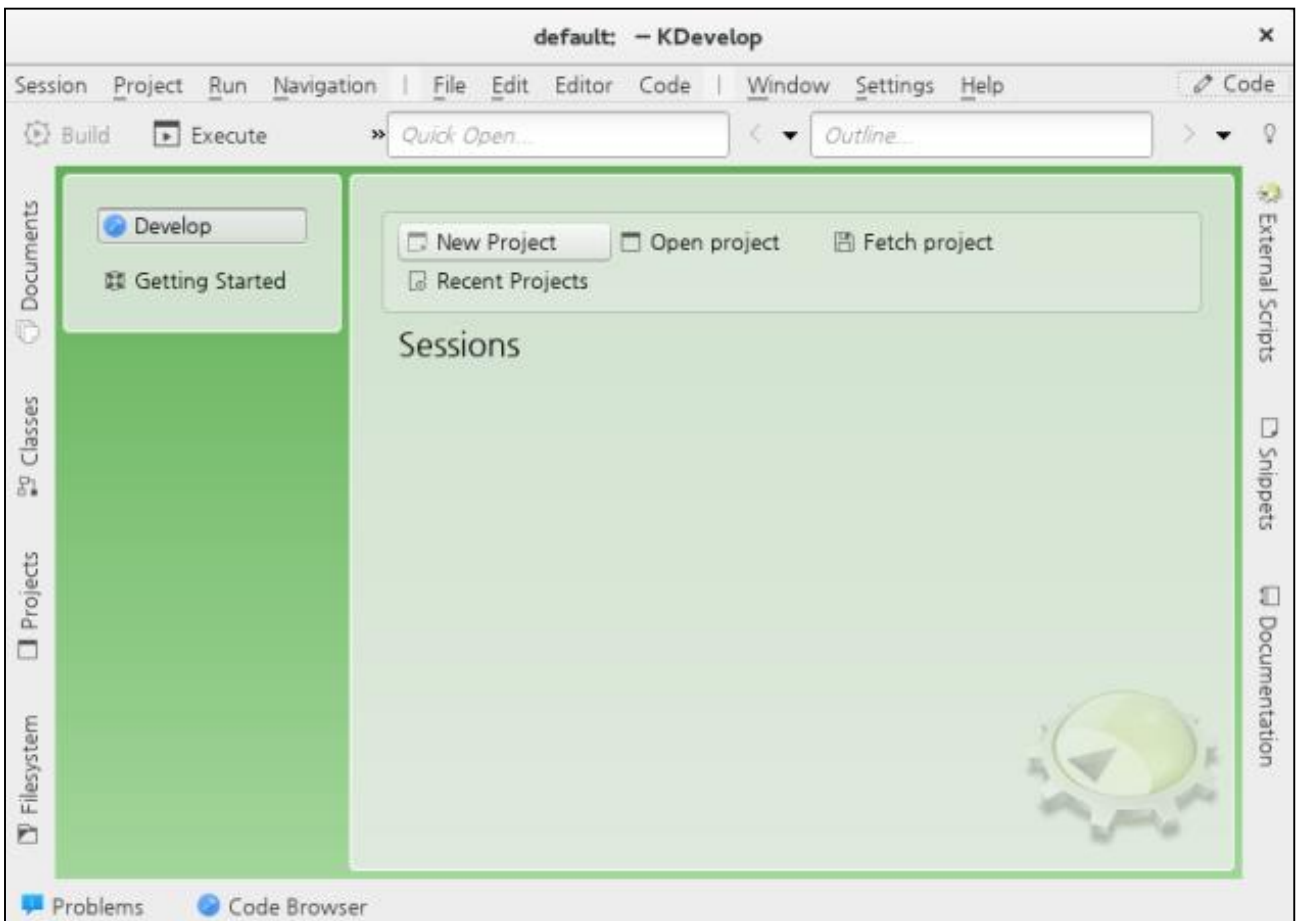
## 프로그래밍 코딩 도구

### 2 GUI 프로그래밍 코딩 도구

#### 4) KDevelop

KDevelop

- KDE 데스크톱 환경에 적합
- 다양한 언어 지원
  - C/C++, Python, Java, Pascal, Fortran, Ada, Perl, Php, Ruby, Sql, Bash Script 등
- 윈도우 운영체제의 Visual Studio와 인터페이스 유사
- 명령어 실행 시, 해당 패키지 설치



KDevelop 실행화면

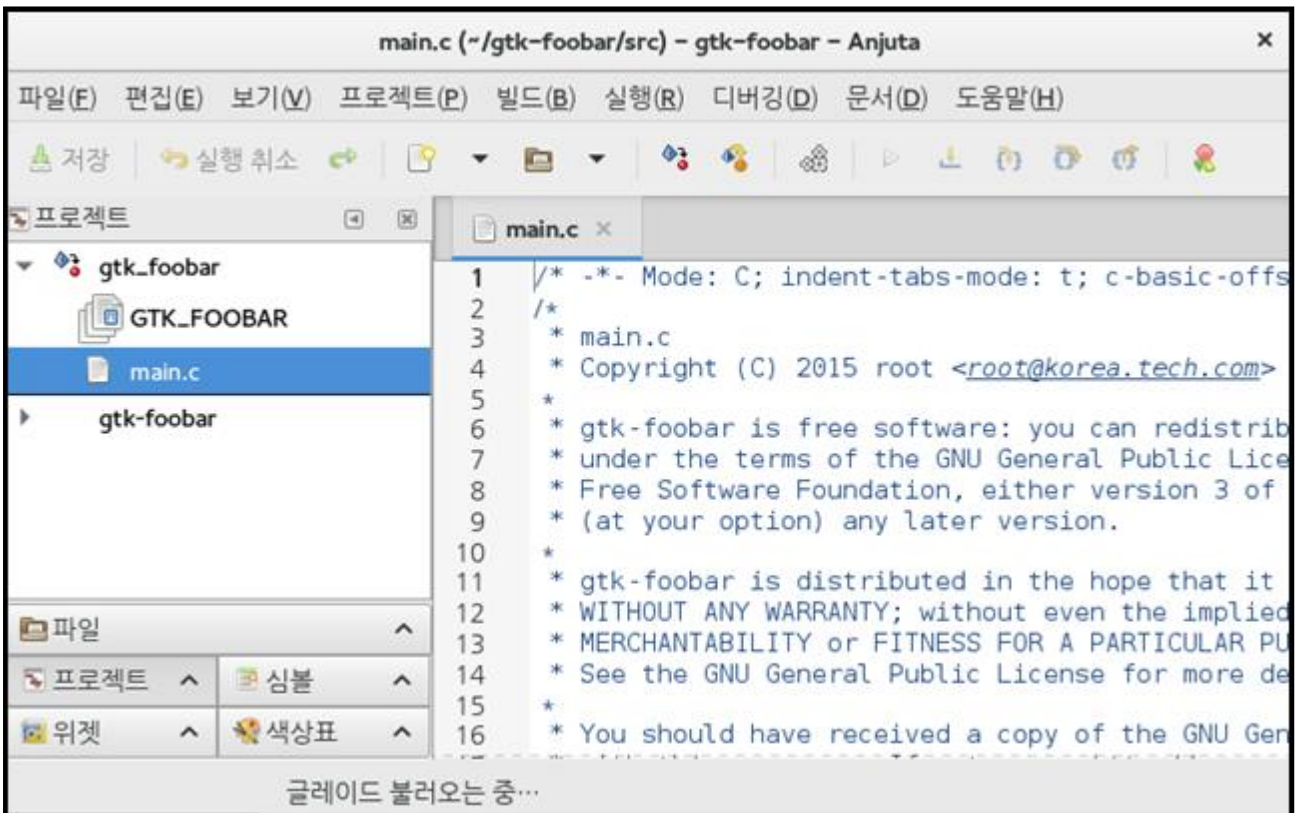
## 프로그래밍 코딩 도구

### 2 GUI 프로그래밍 코딩 도구

#### 5) Anjuta

##### Anjuta

- GNOME 데스크톱 환경에 적합
- 다양한 언어지원
  - C/C++, Java, Python, Vala
- 형상관리 툴인 CVS와 연동
- 명령어 실행 시 설치



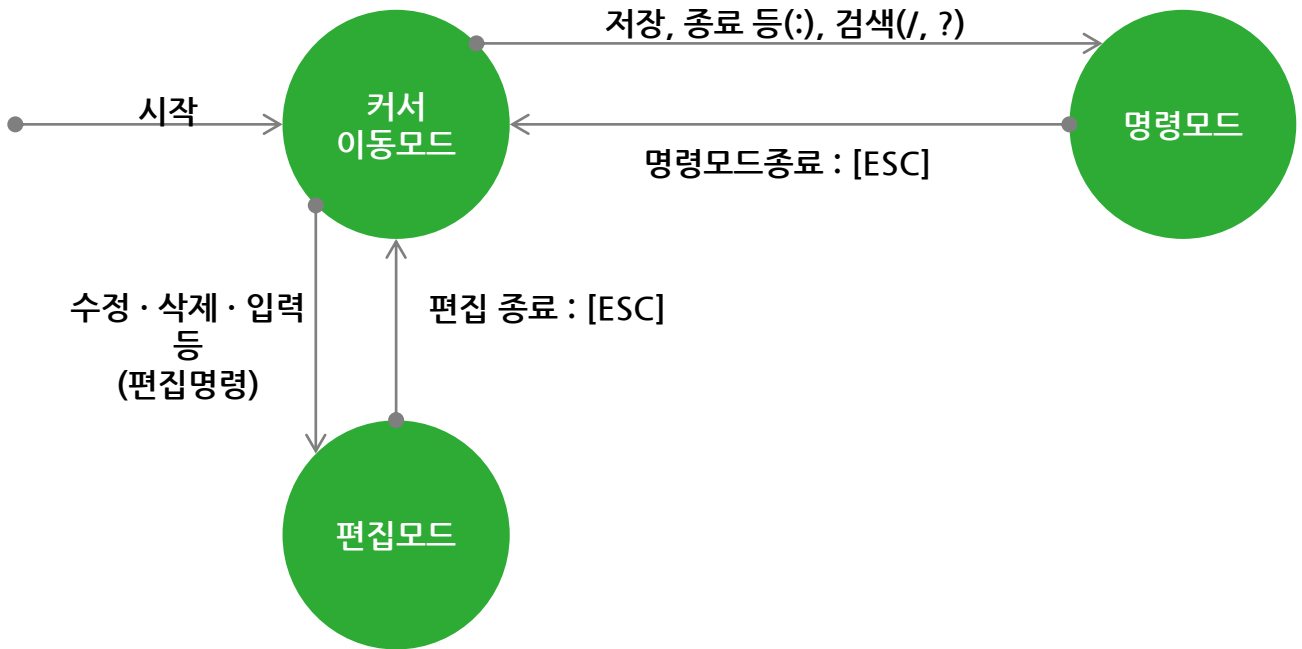
#### Anjuta 실행화면



## Vi 편집기와 gcc 컴파일러

### 1 vi 편집기

#### 1) vi 편집기



## Vi 편집기와 gcc 컴파일러

### 1 vi 편집기

#### 2) 입력 명령어

명령	동작
a	커서 뒤 칸으로 입력
A	커서 위치 라인의 마지막 칸으로 입력
i 또는 Insert	커서 앞칸으로 입력
I	커서 위치 라인의 첫 칸에 입력
o	커서 있는 라인의 아래로 라인을 추가하면서 입력
O	커서 있는 라인의 위로 라인을 추가하면서 입력
[문자 수]s	커서 위치의 문자를 대체하면서 새로운 문자를 입력
[라인 수]S	커서 위치의 라인 전체를 대체하면서 새로운 문자들을 입력
r	커서 위치의 문자를 덮어쓰면서 새로운 문자로 입력
R	커서 위치부터 문자열을 덮어쓰면서 새로운 문자를 입력

[ESC] 키를 입력할 때까지 문자들을 입력할 수 있음



## Vi 편집기와 gcc 컴파일러

### 1 vi 편집기

#### 3) 커서이동 명령어

명령	동작
PgDn, ^f(Ctrl + f)	한 페이지 아래로 이동
PgUp, ^b(Ctrl + b)	한 페이지 위로 이동
Home, 0	커서가 있는 라인의 첫 번째 컬럼으로 이동
End, \$	커서가 있는 라인의 마지막 컬럼으로 이동
w	한 단어씩 앞으로 이동(스페이스 기준)
b	한 단어씩 뒤로 이동



## Vi 편집기와 gcc 컴파일러

### 1 vi 편집기

#### 4) 삭제 명령

명령	동작
[문자 수]x [문자 수]Delete	현재 커서가 있는 <b>한 문자</b> 를 삭제, '3x'는 3개 문자 삭제
[문자 수]X	현재 커서가 있는 <b>앞의 한 문자</b> 를 삭제, '3X'는 앞에 3개 문자 삭제
[라인 수]dd	현재 커서가 있는 <b>한 라인</b> 삭제, '3dd'는 3줄 삭제
D	현재 커서가 있는 위치에서 <b>뒤에 있는 모든 문자</b> 를 삭제

## Vi 편집기와 gcc 컴파일러

### 1 vi 편집기

#### 5) 복사 및 검색 명령

구분	명령	동작
복사 명령	[라인수]yy	■ <b>한 줄 복사</b> , '3yy'는 3줄 복사
	[라인수]p	■ <b>붙여넣기</b> , '3p'는 3번 연속 붙여넣기
검색 명령	/문자열	■ <b>문서의 아래</b> 로 찾고자 하는 문자열을 검색하여 커서를 이동시킴 ■ 현재 커서 위치에서 아래로 검색함 ■ 계속 검색하기 위해서는 'n' 또는 'N(검색방향 반대)'을 입력함
	?문자열	■ <b>문서의 위</b> 로 찾고자 하는 문자열을 검색하여 커서를 이동시킴 ■ 현재 커서 위치에서 위로 검색 ■ 계속 검색하기 위해서는 'n' 또는 'N(검색방향 반대)'을 입력

## Vi 편집기와 gcc 컴파일러

### 1 vi 편집기

#### 6) 명령 취소 및 반복 명령

명령	동작
u	바로 전에 했던 명령을 취소함(undo 기능)
.	바로 전에 했던 명령을 <b>반복</b> 함



## Vi 편집기와 gcc 컴파일러

### 1 vi 편집기

#### 7) 저장 및 종료 명령

명령	동작
:q	저장하지 않고 종료, 만약 수정되었다면 종료가 되지 않음
:q!	저장하지 않고 종료, 만약 수정되었더라도 강제로 종료됨
:wq	저장하고 종료
:wq!	저장하고 강제 종료

## Vi 편집기와 gcc 컴파일러

### 2 gcc 컴파일러

#### 1) gcc(GNU C Compiler)컴파일러 개요

##### gcc(GNU C Compiler)

- GNU에서 만든 C 컴파일러
- 지원 CPU 아키텍처
- ARM, DEC, AVR, i386, PPC, SPARC, M68xx등
- gcc 컴파일 가능 언어
- C, Fortran, Ada, Java, Objective-C
- gcc 명령 실행 시 설치

##### GCC(GNU Compiler Collection)

- 다양한 컴파일러를 포함하는 컴파일러 묶음
- 예 gcc, c++, Java, Ada, Fortran, Objective-C



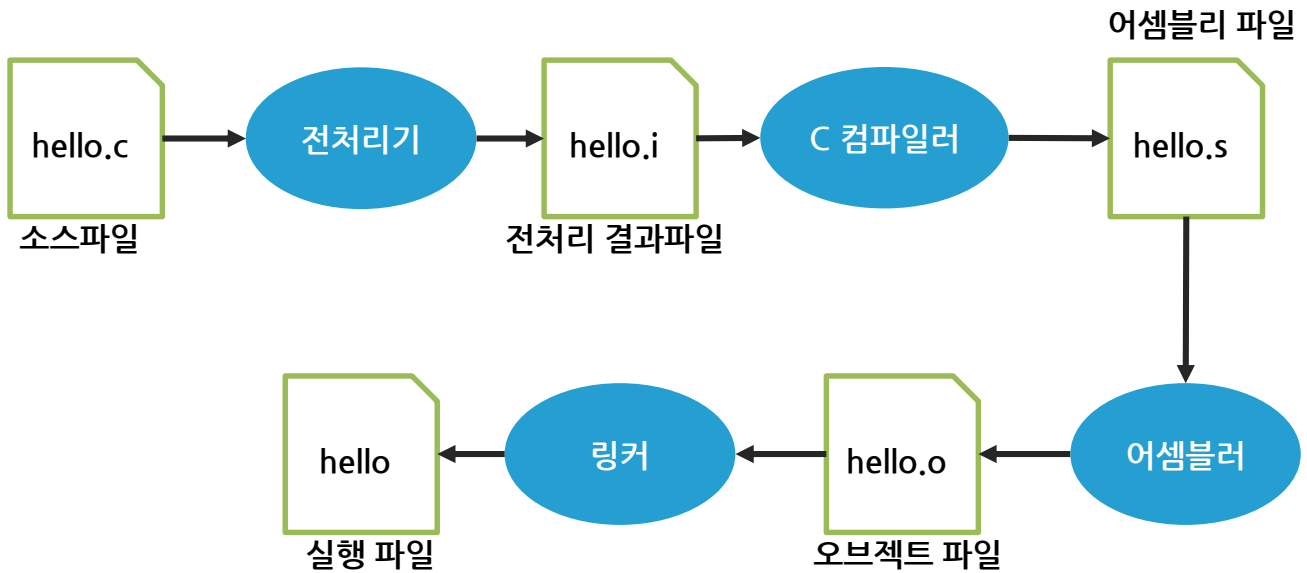
GCC와 gcc는 다른 의미로 사용



## Vi 편집기와 gcc 컴파일러

### 2) gcc 컴파일러

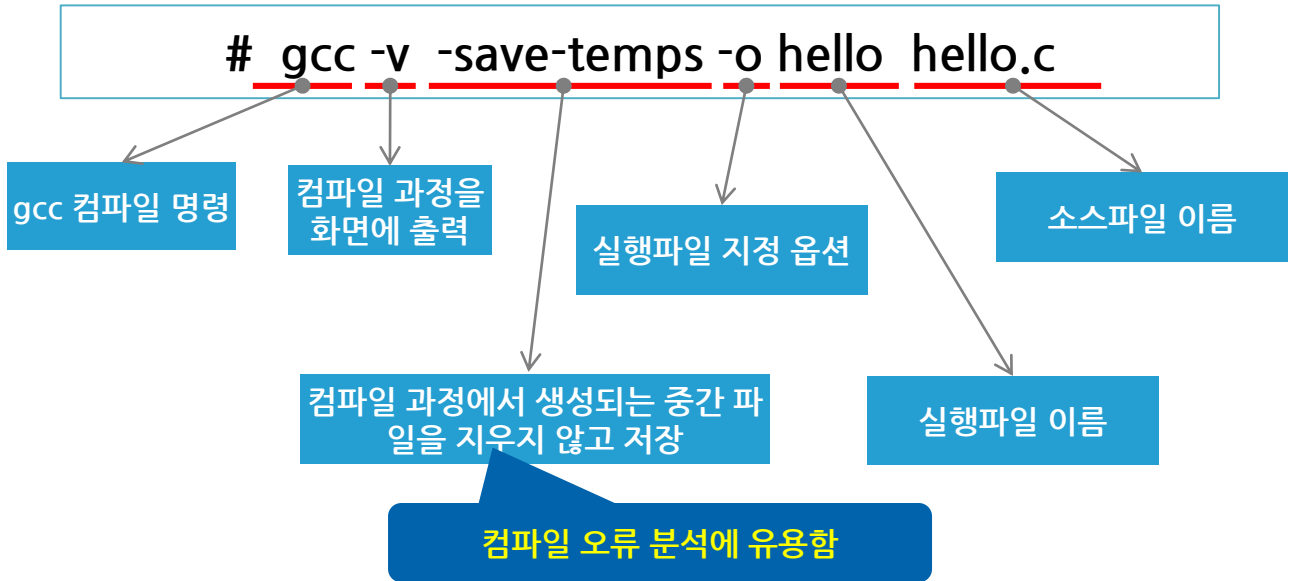
#### 2) gcc 컴파일 과정



## Vi 편집기와 gcc 컴파일러

### 2) gcc 컴파일러

#### 3) gcc 컴파일 명령



## 프로그래밍 코딩 도구

---

### + 텍스트 프로그래밍 코딩 도구

- vi, vim

### + GUI 프로그래밍 코딩 도구

- Emacs, Eclipse, gedit, KDevelop, Anjuta

## vi 편집기와 gcc 컴파일러

---

### + vi 편집기

- 실행모드 : 커서이동모드, 명령모드, 편집모드
- 입력명령 : a, A, i, I, o, O, s, S, r, R
- 커서이동 명령 : 화살표, PgUp, PgDn, Ctrl+f, Ctrl+b, w, b
- 삭제명령 : Delete, x, X, D, dd
- 복사 및 붙여넣기 명령 : yy, p
- 검색명령 : /, ?, n, N
- 명령 취소 및 반복 : u, .
- 저장 및 종료 명령 : :q, :q!, :wq, :wq!

### + gcc 컴파일러

- GNU에서 만든 C 컴파일러
- 소스파일(\*.c) - 전처리기(\*.i) - 컴파일러(\*.s) - 어셈블러(\*.o) - 링커 - 실행파일
- 컴파일 명령 : # gcc -v -save-temps -o hello hello.c



### + vi 편집기 명령

기능	명령
입력 명령	a, A, i, I, o, O, s, S, r, R
커서이동 명령	화살표, PgUp, PgDn, Ctrl+f, Ctrl+b, w, b
삭제 명령	Delete, x, X, D, dd
복사 및 붙여넣기 명령	yy, p
검색명령	/, ?, n, N
명령 취소 및 반복	u, .
저장 및 종료 명령	:q, :q!, :wq, :wq!