

# 2019\_04\_13 3차시

## GCC 컴파일 과정 알아보기

### 1. GCC란?

1. GNU 컴파일러 모음은 GNU 프로젝트의 일환으로 개발되어 널리 쓰이고 있는 컴파일러이다.

```
compilation terminated.
m0nday@trust ~/gongup1ban/gcc $ gcc --help
Usage: gcc [options] file...
Options:
  -pass-exit-codes      Exit with highest error code from a phase
  --help                Display this information
  --target-help          Display target specific command line options
  --help={common|optimizers|params|target|warnings|[^]{joined|separate|undocumented}}[,...]
                        Display specific types of command line options
  (Use '-v --help' to display command line options of sub-processes)
  --version             Display compiler version information
  -dumpspecs             Display all of the built in spec strings
  -dumpversion           Display the version of the compiler
  -dumpmachine           Display the compiler's target processor
  -print-search-dirs     Display the directories in the compiler's search path
  -print-libgcc-file-name
                        Display the name of the compiler's companion library
  -print-file-name=<lib>
                        Display the full path to library <lib>
  -print-prog-name=<prog>
                        Display the full path to compiler component <prog>
  -print-multiarch       Display the target's normalized GNU triplet, used as
                        a component in the library path
  -print-multi-directory
                        Display the root directory for versions of libgcc
  -print-multi-lib       Display the mapping between command line options and
                        multiple library search directories
  -print-multi-os-directory
                        Display the relative path to OS libraries
  -print-sysroot         Display the target libraries directory
  -print-sysroot-headers-suffix
                        Display the sysroot suffix used to find headers
  -Wa,<options>          Pass comma-separated <options> on to the assembler
  -Wp,<options>          Pass comma-separated <options> on to the preprocessor
  -Wl,<options>          Pass comma-separated <options> on to the linker
  -Xassembler <arg>     Pass <arg> on to the assembler
  -Xpreprocessor <arg>  Pass <arg> on to the preprocessor
  -Xlinker <arg>         Pass <arg> on to the linker
  -save-temps            Do not delete intermediate files
  -save-temps=<arg>     Do not delete intermediate files
  -no-canonical-prefixes
                        Do not canonicalize paths when building relative
                        prefixes to other gcc components
  -pipe                 Use pipes rather than intermediate files
  -time                  Time the execution of each subprocess
  -specs=<file>          Override built-in specs with the contents of <file>
  -std=<standard>        Assume that the input sources are for <standard>
  --sysroot=<directory>
                        Use <directory> as the root directory for headers
                        and libraries
  -B <directory>        Add <directory> to the compiler's search paths
```

### 2. GCC 컴파일 옵션

1. -Wall : 모든 모호한 코딩에 대해서 경고를 보내는 옵션
2. -W : 합법적이지만 모호한 코딩에 대해서 경고를 보내는 옵션
3. -W -Wall : 아주 사소한 모호성에 대해서도 경고가 발생
4. -O2 : 최적화 레벨 2로 설정. (대부분의 최적화를 시도)
5. -E : 전처리 과정의 결과를 화면에 보이는 옵션
6. -S : cc1으로 전처리된 파일을 어셈블리 파일로 컴파일까지만 수행하고 멈춘다.
7. -v : gcc가 컴파일을 어떤 식으로 수행하는지를 화면에 출력한다.

8. -c 옵션 : as에 의한 어셈블까지만 수행하고 링크는 수행하지 않는다.

### 3. gcc 컴파일 과정

1. gcc 파일명(\*.c) : Default로 out 파일이 생성된다. (ex a.out)
2. gcc -c 파일명(\*.c) : 오브젝트 파일을 생성한다.
3. gcc -c 오브젝트\_파일명(\*.o) 파일명(\*.c)
4. gcc -o 실행파일명(\*.out) 오브젝트\_파일명(\*.o)

### 4. 실제 컴파일 해보기

1. 별짓기 프로그래밍

```
m0nday@trust ~/gongup1ban/gcc $ cat example.c
#include <stdio.h>

int main() {

for (int i = 0; i < 10; i++) {
1.         for (int j = 0; j < i; j++) {
                printf("*");
            }
        printf("\n");
    }
    return 0;
}
```

2. 컴파일

```
m0nday@trust ~/gongup1ban/gcc $ gcc -o example example.c
1. m0nday@trust ~/gongup1ban/gcc $ ls
example  example.c
```

3. 실행하기

```
m0nday@trust ~/gongup1ban/gcc $ ./example

*
**
***
1. ****
*****
*****
*****
*****
*****
*****
```