GCC C/C++ compiler

CONCEPTS

gcc drives four stages: preprocess, compile, assemble, link. -E, -S and -c stop after the first three.

x.c source
↓ -E
x.i preprocessed
↓ -S
x.s assembly
↓ -c
x.o object
↓ link
a.out executable

Each stage has a flag that stops after it

Put libraries (-l) after the files that use them: the linker resolves symbols left to right.

COMPILE AND LINK

`gcc <file.c> -o <out>`
Build an executable

`gcc -c <file.c>`
Compile to an object file

`gcc <a.o> <b.o> -o <out>`
Link objects together

`gcc <a.c> <b.c> -o <out>`
Compile several files at once

LANGUAGE STANDARD

`gcc -std=c17 <file.c>`
Pick the C standard

`g++ -std=c++20 <file.cpp>`
Pick the C++ standard

WARNINGS

`gcc -Wall <file.c>`
Common warnings

`gcc -Wextra <file.c>`
Extra warnings on top of -Wall

`gcc -Werror <file.c>`
Treat warnings as errors

OPTIMIZATION

`gcc -O0 <file.c>`
No optimization (default)

OPTIMIZATION (cont.)

`gcc -O2 <file.c>`
Standard release optimization

DEBUG AND PROFILE

`gcc -g <file.c>`
Add debug info for gdb

SANITIZERS

`gcc -fsanitize=address -g <file.c>`
AddressSanitizer: memory errors

`gcc -fsanitize=undefined -g <file.c>`
UBSan: undefined behavior

`gcc -fsanitize=address,undefined -g <file.c>`
Combine ASan and UBSan

INCLUDE AND LINK PATHS

`gcc -I<dir> <file.c>`
Add a header search dir

`gcc -L<dir> <file.c> -l<name>`
Add a library search dir

`gcc <file.c> -l<name>`
Link libname.so or libname.a

`gcc <file.c> -lm`
Link the math library

MACROS

`gcc -D<name> <file.c>`
Define a macro as 1

`gcc -D<name>=<value> <file.c>`
Define a macro with a value

DEPENDENCIES AND LIBRARIES

`gcc -MMD -MP -c <file.c>`
Write .d dependency file with build

`gcc -fPIC -c <file.c>`
Position-independent object

`gcc -shared -o lib<name>.so <a.o>`
Build a shared library

INSPECT THE COMPILER

`gcc --version`
Show the GCC version