g++ compilation flags

Adam Graliński

C++ FFFE, February 2021

Flags

```
g++ [-c|-S|-E] [-std=standard]
    [-g] [-pg] [-Olevel]
    [-Wwarn...] [-Wpedantic]
    [-Idir...] [-Ldir...]
    [-Dmacro[=defn]...] [-Umacro]
    [-foption...] [-mmachine-option...]
    [-o outfile] [@file] infile...
```

-c, -S, -E : stop after build phase

`-c`	stop after compilation & assembly — do not link
`-S`	stop after compilation — do not assemble
`-E`	stop after preprocessing step — do not compile

-g, -p: generate extra information

`-g`	emit extra information for a debugger
`-ggdb`	(for a gdb debugger)
`-p`	emit extra information for *prof* profiler
`-pg`	emit extra information for a *gprof* profiler

-O: fine-tune the optimization

-O0	optimize for compilation time (this implies **no code optimizations**)
-O1	enable moderate optimization
-O2	enable full optimization
-O3	enable full+aggressive optimization

Flags enabled by -Olevel

- -O1
  -fauto-inc-dec -fbranch-count-reg -fcombine-stack-adjustments -fcompare-elim -fcprop-registers -fdce -fdefer-pop -fdelayed-branch -fdse -fforward-propagate -fguess-branch-probability -fif-conversion -fif-conversion2 -finline-functions-called-once -fipa-modref -fipa-profile -fipa-pure-const -fipa-reference -fipa-reference-addressable -fmerge-constants -fmove-loop-invariants -fomit-frame-pointer -freorder-blocks -fshrink-wrap -fshrink-wrap-separate -fsplit-wide-types -fssa-backprop -fssa-phiopt -ftree-bit-ccp -ftree-ccp -ftree-ch -ftree-coalesce-vars -ftree-copy-prop -ftree-dce -ftree-dominator-opts -ftree-dse -ftree-forwprop -ftree-fre -ftree-phiprop -ftree-pta -ftree-scev-cprop -ftree-sink -ftree-slsr -ftree-sra -ftree-ter -funit-at-a-time

- -O2
  -falign-functions -falign-jumps -falign-labels -falign-loops -fcaller-saves -fcode-hoisting -fcrossjumping -fcse-follow-jumps -fcse-skip-blocks -fdelete-null-pointer-checks -fdevirtualize -fdevirtualize-speculatively -fexpensive-optimizations -ffinite-loops -fgcse -fgcse-lm -fhoist-adjacent-loads -finline-functions -finline-small-functions -findirect-inlining -fipa-bit-cp -fipa-cp -fipa-icf -fipa-ra -fipa-sra -fipa-vrp -fisolate-erroneous-paths-dereference -flra-remat -foptimize-sibling-calls -foptimize-strlen -fpartial-inlining -fpeephole2 -freorder-blocks-algorithm=stc -freorder-blocks-and-partition -freorder-functions -frerun-cse-after-loop -fschedule-insns -fschedule-insns2 -fsched-interblock -fsched-spec -fstore-merging -fstrict-aliasing -fthread-jumps -ftree-builtin-call-dce -ftree-pre -ftree-switch-conversion -ftree-tail-merge -ftree-vrp

- -O3
  -fgcse-after-reload -fipa-cp-clone -floop-interchange -floop-unroll-and-jam -fpeel-loops -fpredictive-commoning -fsplit-loops -fsplit-paths -ftree-loop-distribution -ftree-loop-vectorize -ftree-partial-pre -ftree-slp-vectorize -funswitch-loops -fvect-cost-model -fvect-cost-model=dynamic -fversion-loops-for-strides

- -Os
  -O2 **except**: -falign-functions -falign-jumps -falign-labels -falign-loops -fprefetch-loop-arrays -freorder-blocks-algorithm=stc

-I, -L: search these directories

`-I`*directory*	add *directory* to a search-list for header files during preprocessing
`-L`*directory*	add *directory* to a search-list for libraries (-l)

-D, -U: (un-)define macros

`-D`**name**	define a **name** macro it will be processed as if it was defined in a translation unit via a `#define` directive
`-U`**name**	cancel any previous definition of **name** macro

@file: options' file

@**file**	read command-line options for gcc from **file**.
	The options are inserted in place of @**file** option.
	file structure: space-separated commands,
	can contain additional @**file** directives.

Flags, revisited

```
g++ [-c|-S|-E] [-std=standard]
    [-g] [-pg] [-Olevel]
    [-Wwarn...] [-Wpedantic]
    [-Idir...] [-Ldir...]
    [-Dmacro[=defn]...] [-Umacro]
    [-foption...] [-mmachine-option...]
    [-o outfile] [@file] infile...
```

Thank you!