

gcc -O option flag

Set the compiler's optimization level.

| option | optimization level | execution time | code size | memory usage | compile time |
|-----------|--|----------------|-----------|--------------|--------------|
| -O0 | optimization for compilation time (default) | + | + | - | - |
| -O1 or -O | optimization for code size and execution time | - | - | + | + |
| -O2 | optimization more for code size and execution time | -- | | + | ++ |
| -O3 | optimization more for code size and execution time | --- | | + | +++ |
| -Os | optimization for code size | | -- | | ++ |
| -Ofast | O3 with fast none accurate math calculations | --- | | + | +++ |

+increase ++increase more +++increase even more -reduce --reduce more ---reduce even more

<https://linux.die.net/man/1/gcc>

Options That Control Optimization

These options control various sorts of optimizations.

Without any optimization option, the compiler's goal is to reduce the cost of compilation and to make debugging produce the expected results. Statements are independent: if you stop the program with a breakpoint between statements, you can then assign a new value to any variable or change the program counter to any other statement in the function and get exactly the results you would expect from the source code.

Turning on optimization flags makes the compiler attempt to improve the performance and/or code size at the expense of compilation time and possibly the ability to debug the program.

The compiler performs optimization based on the knowledge it has of the program. Compiling multiple files at once to a single output file mode allows the compiler to use information gained from all of the files when compiling each of them.

Not all optimizations are controlled directly by a flag. Only optimizations that have a flag are listed.

-O

-O1

Optimize. Optimizing compilation takes somewhat more time, and a lot more memory for a large function.

With **-O**, the compiler tries to reduce code size and execution time, without performing any optimizations that take a great deal of compilation time.

-O turns on the following optimization flags:

-fauto-inc-dec -fcprop-registers -fdce -fdefer-pop -fdelayed-branch -fdse -fguess-branch-probability -fif-conversion2 -fif-conversion -finline-small-functions -fipa-pure-const -fipa-reference -fmerge-constants -fsplit-wide-types -ftree-builtin-call-dce -ftree-ccp -ftree-ch -ftree-copyrename -ftree-dce -ftree-dominator-opts -ftree-dse -ftree-fre -ftree-sra -ftree-ter -funit-at-a-time

-O also turns on **-fomit-frame-pointer** on machines where doing so does not interfere with debugging.

-O2

Optimize even more. GCC performs nearly all supported optimizations that do not involve a space-speed tradeoff. As compared to **-O**, this option increases both compilation time and the performance of the generated code.

-O2 turns on all optimization flags specified by **-O**. It also turns on the following optimization flags: **-fthread-jumps -falign-functions -falign-jumps -falign-loops -falign-labels -fcaller-saves -fcrossjumping -fcse-follow-jumps -fcse-skip-blocks -fdelete-null-pointer-checks -fexpensive-optimizations -fgcse -fgcse-lm -findirect-inlining -foptimize-sibling-calls -fpeephole2 -fregmove -freorder-blocks -freorder-functions -frerun-cse-after-loop -fsched-interblock -fsched-spec -fschedule-insns -fschedule-insns2 -fstrict-aliasing -fstrict-overflow -ftree-switch-conversion -ftree-pre -ftree-vrp**

Please note the warning under **-fgcse** about invoking **-O2** on programs that use computed gotos.

-O3

Optimize yet more. **-O3** turns on all optimizations specified by **-O2** and also turns on the **-finline-functions**, **-funswitch-loops**, **-fpredictive-commoning**, **-fgcse-after-reload**, **-ftree-vectorize** and **-fipa-cp-clone** options.

-O0

Reduce compilation time and make debugging produce the expected results. This is the default.

-Os

Optimize for size. **-Os** enables all **-O2** optimizations that do not typically increase code size. It also performs further optimizations designed to reduce code size.

-Os disables the following optimization flags: **-falign-functions -falign-jumps -falign-loops -falign-labels -freorder-blocks -freorder-blocks-and-partition -fprefetch-loop-arrays -ftree-vect-loop-version**

If you use multiple **-O** options, with or without level numbers, the last such option is the one that is effective.

Options of the form **-fflag** specify machine-independent flags. Most flags have both positive and negative forms; the negative form of **-ffoo** would be **-fno-foo**. In the table below, only one of the forms is listed---the one you typically will use. You can figure out the other form by either removing **no-** or adding it.

The following options control specific optimizations. They are either activated by **-O** options or are related to ones that are. You can use the following flags in the rare cases when "fine-tuning" of optimizations to be performed is desired.

https://wiki.gentoo.org/wiki/GCC_optimization/ru

-O

Следующая по списку - переменная **-O**. Она управляет всем уровнем оптимизации.

Изменение этой переменной приводит к тому, что компиляция кода занимает больше времени, и сможет занять гораздо больше памяти, особенно когда уровень оптимизации увеличен.

Существует семь видов настроек переменной **-O**: **-O0**, **-O1**, **-O2**, **-O3**, **-Os**, **-Og** и **-Ofast**. Используйте только одну из них в `/etc/portage/make.conf`.

За исключением **-O0**, каждая из настроек с префиксом **-O** активирует несколько дополнительных флагов, поэтому удостоверьтесь, что Вы прочитали главу руководства GCC по [параметрам оптимизации](#) для изучения того, какие флаги активируются на каждом уровне с приставкой **-O**, также как и некоторые из объяснений того, что они делают.

Давайте исследуем каждый уровень оптимизации:

- **-O0**: Этот уровень (буква "O" и ноль за ней) отключает оптимизацию полностью и является уровнем по умолчанию, если никакого уровня с префиксом **-O** не указано в переменных `CFLAGS` или `CXXFLAGS`. Это сокращает время компиляции и может улучшить данные для отладки, но некоторые приложения не будут работать должным образом без оптимизации. Эта опция не рекомендуется, за исключением использования в целях отладки.
- **-O1**: Это наиболее простой уровень оптимизации. Компилятор попытается сгенерировать быстрый, занимающий меньше объема код, без затрачивая наибольшего времени компиляции. Он достаточно простой, но должен всегда выполнять свою работу.

- `-O2` : Шаг вперед от `-O1` . *Рекомендуемый* уровень оптимизации, до тех пор пока не понадобится что-то особенное. `-O2` активирует несколько дополнительных флагов вдобавок к флагам, активированных `-O1` . С параметром `-O2` , компилятор попытается увеличить производительность кода без нарушения размера, и без затрачивания большого количества времени компиляции.
- `-O3` : Это наибольший возможный уровень оптимизации. Включает оптимизации, являющейся дорогостоящей с точки зрения времени компиляции и потребления памяти. Компиляция с `-O3` не является гарантированным способом повышения производительности, и на самом деле во многих случаях может привести к замедлению системы из-за больших двоичных файлов и увеличения потребления памяти. `-O3` известен также тем, что ломает несколько пакетов. Использование `-O3` не рекомендуется.
- `-Os` : На этом уровне код будет оптимизирован по объему. Он активирует все параметры `-O2` , которые не приводят к увеличению размера генерируемого кода. Он может быть полезным на компьютерах, которые обладают чрезвычайно ограниченным пространством жесткого диска и/или процессоры с небольшим размером кэша.
- `-Og` : В GCC 4.8 был введен новый общий уровень оптимизации `-Og` . Он удовлетворяет потребность в быстрой компиляции и имеет превосходные возможности для отладки, обеспечивая при этом приемлемый уровень производительности во время выполнения. Общий опыт разработки должен быть лучше, чем с уровнем оптимизации по умолчанию `-O0` . Обратите внимание, что `-Og` не означает `-g` , он просто отключает оптимизацию кода, которая может помешать отладке.
- `-Ofast` : Новое в GCC 4.7, состоит из `-O3` плюс `-ffast-math` , `-fno-protect-parens` , и `-fstack-arrays` . Этот параметр нарушает строгое соответствие стандарту, и не рекомендуется для использования.

Как упомянуто ранее, параметр `-O2` - рекомендуемый уровень оптимизации. Если компиляция пакета выдает сообщение об ошибке и не используется параметр `-O2` , то попробуйте перекомпилировать с этой опцией. В качестве выхода, попробуйте установить переменные `CFLAGS` и `CXXFLAGS` на наименьший уровень оптимизации, такой как `-O1` , или даже `-O0 -g2 -g gdb` (для сообщения об ошибках и проверки возможных проблем).