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.

За исключением -O0, каждая из настроек с префиксом -O активирует несколько дополнительных флагов.

-O0: Этот уровень 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 затрачивания 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. На этом уровне могут быть использованы SSE (потоковое SIMD-расширение процессора)  [SIMD](https://ru.wikipedia.org/wiki/SIMD)- ( *Single Instruction, Multiple Data*, Одна инструкция — множество данных) : подобие распараллеливания) и AVX (доп инструкции : увеличение размеров регистра, отсутствие требования выравнивания операндов в памяти тд), но YMM-регистры не будут использоваться пока не будет включена опция -ftree-vectorize. -ftree-vectorize - опция оптимизации (по умолчанию при -O3 и -Ofast), которая по возможности пытается векторизовать циклы, используя выбранную ISA. *Причина, по которой она не включена при -O2 - она не всегда улучшает код, может также сделать его медленнее, и обычно делает код больше; зависит от конкретного цикла и т.д.*

-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. Также включает -ftree-vectorize, и циклы в коде векторизируются и используют регистры AVX YMM.

-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 перекопилировать с этой опцией. В качестве выхода, попробуйте установить переменные CFLAGS и CXXFLAGS на наименьший уровень оптимизации, такой как -O1, или даже -O0 -g2 -ggdb (для сообщения об ошибках и проверки возможных проблем).

-O0 Отключает оптимизацию

-О1 Оптимизация по размеру. Не использует методов оптимизации, которые могут увеличить размер кода. Создает в большинстве случаев самый маленький размер кода.

-O2 или -O Максимизация скорости. Как правило, создает более быстрый код, чем -O1. Эта опция устанавливается по умолчанию для оптимизации, если не указана -g

-O3 Задействует методы оптимизации из -O2 и, дополнительно, более агрессивные методы оптимизации циклов и доступа к памяти, такие как подстановка скаляров, раскрутка циклов, подстановка кода для избегания ветвлений, блокирование циклов для обеспечения более эффективного использования кэш-памяти и, только на системах архитектуры IA-64, дополнительная подготовка данных. Данная опция особенно рекомендуется для приложений, где есть циклы, которые активно используют вычисления с плавающей точкой или обрабатывают большие порции данных. Эти агрессивные методы оптимизации могут в ряде случаев, и замедлить работу приложений других типов по сравнению с использованием -O2.