

GCC编译器选项及优化提示 PDF转换可能丢失图片或格式，
建议阅读原文

https://www.100test.com/kao_ti2020/253/2021_2022_GCC_E7_BC_96_E8_AF_91_E5_c103_253005.htm 大多数程序和库在编译时默认的优化级别是"2"(使用gcc选项:"-O2")并且在Intel/AMD平台上默认按照i386处理器来编译。如果你只想让编译出来的程序运行在特定的平台上，就需要执行更高级的编译器优化选项，以产生只能运行于特定平台的代码。一种方法是修改每个源码包中的Makefile文件，在其中寻找CFLAGS和CXXFLAGS变量(C和C++编译器的编译选项)并修改它的值。一些源码包比如binutils, gcc, glibc等等，在每个子文件夹中都有Makefile文件，这样修改起来就太累了！另一种简易做法是设置CFLAGS和CXXFLAGS环境变量。大多数configure脚本会使用这两个环境变量代替Makefile文件中的值。但是少数configure脚本并不这样做，他们必须需要手动编辑才行。为了设置CFLAGS和CXXFLAGS环境变量，你可以在bash中执行如下命令(也可以写进.bashrc以成为默认值)：
export CFLAGS="-O3 -march=" & CXXFLAGS=\$CFLAGS这是一个确保能够在几乎所有平台上都能正常工作的最小设置。
"-march"选项表示为特定的cpu类型编译二进制代码(不能在更低级别的cpu上运行)，Intel通常是：pentium2, pentium3, pentium3m, pentium4, pentium4m, pentium-m, prescott, nocona说明：pentium3m/pentium4m是笔记本用的移动P3/P4；pentium-m是迅驰I/II代笔记本的cpu；prescott是带SSE3的P4(以滚烫到可以煎鸡蛋而闻名)；nocona则是最新的带有EMT64(64位)的P4(同样可以煎鸡蛋)AMD通常是：k6, k6-2,

k6-3, athlon, athlon-tbird, athlon-xp, athlon-mp, opteron, athlon64, athlon-fx用AMD的一般都是DIYer，就不必解释了吧。如果编译时没有抱怨"segmentation fault, core dumped"，那么你设定的"-O"优化参数一般就没什么问题。否则请降低优化级别("-O3" -> "-O2" -> "-O1" -> 取消)。个人意见：服务器使用"-O2"就可以了，它是最安全的优化参数(集合)；桌面可以使用"-O3"；不鼓励使用过多的自定义优化选项，其实他们之间没什么明显的速度差异(有时"-O3"反而更慢)。编译器对硬件非常敏感，特别是在使用较高的优化级别的时候，一丁点的内存错误都可能导致致命的失败。所以在编译时请千万不要超频你的电脑(我编译关键程序时总是先降频然的)。注意：选项的顺序很重要，如果有两个选项互相冲突，则以后一个为准。比如"-O3"将打开-finline-functions选项，但是可以用"-O3 -fno-inline-functions"既使用-O3的功能又关闭函数内嵌功能。声明，"-O2"已经启用绝大多数安全的优化选项了，所以其实你不必对那一堆选项发愁。先说说"-O3"在"-O2"基础上增加的几项，你可以按需添加(还算比较安全)

：[gcc-3.4.4]-finline-functions 允许编译器选择某些简单的函数在其被调用处展开-fweb 为每个web结构体分配一个伪寄存器-frename-registers 试图驱除代码中的假依赖关系，这个选项对具有大量寄存器的机器很有效。[gcc-4.0.2]-finline-functions 说明如上-funswitch-loops 将循环体中不改变值的变量移动到循环体之外-fgcse-after-reload **不太明白它的含义**[哪位大侠知道给小弟讲解一下，先行谢过]说完"-O3"再说说在嵌入式系统上常用的"-Os"选项，这个选项其实也很重要，它的含义是对生成的二进制代码进行尺寸上的优化，它打开了所有"-

O2"打开的选项，因此通常认为的"-Os"生成的二进制代码执行效率低的潜在意识是错误的！当然该选项与"-O2"的不同之处在于它在"-O2"的基础上禁止了所有为了对齐而插入的空间，也就是将所有"-falign-*"系列的选项禁用了。这种禁用究竟是否一定降低了代码的执行效率，依据程序的不同而不同，据说某些情况下"-Os"的效率比"-O3"还要高14%！请兄弟们在实践中自己摸索吧...-----

下面选择我认为比较重要的几项简单介绍一下[gcc-3.4.4][注意]这里列出的都是非默认的选项，你只需要添加你所需要的选项即可-w 禁止输出警告消息-Werror 将所有警告转换为错误-Wall 显示所有的警告消息-v 显示编译程序的当前版本号-V 指定gcc将要运行的版本。只有在安装了多个版本gcc的机器上才有效。-ansi 按照ANSI标准编译程序，但并不限制与标准并不冲突的GNU扩展(一般不用该选项)-pedantic 如果要限制代码必须严格符合ISO标准，就在"-ansi"的基础上同时启用这个选项(很少使用)-std= 指定C语言的标准(c89,c99,gnu89)，该选项禁止了GNU C的扩展关键字asm,typeof,inline (一般不用该选项)-static 连接器将忽略动态链接库，同时通过将静态目标文件直接包含到结果目标文件完成对所有引用的解析。-shared 连接器将生成共享目标代码，该共享库可在运行时动态连接到程序形成完整的可执行体。如果使用gcc命令创建共享库作为其输出，该选项可以防止连接器将缺失main()方法视为错误。为了可以正确的工作，应该一致的使用选项"-fpic"以及目标平台选项编译构成同一个库的所有共享目标模块。-shared-libgcc 该选项指定使用共享版本的libgcc，在没有共享版本的libgcc的机器上该选项无效。-specs= gcc驱动程序读

取该文件以确定哪些选项应该传递给那些子进程。该选项可以通过指定配置文件来覆盖默认配置，指定的文件将在默认配置文件读取后进行处理以修改默认配置。-pipe 使用管道而不是临时文件一个阶段到另一个阶段交换输出的方式，可以加快编译速度。建议使用。-o 指定输出文件，对各种输出皆有效。由于只能指定一个文件，所以在产生多个输出文件的情况下不要使用该选项。--help 显示gcc的命令行选项列表；与"-v"一起使用时还将显示gcc调用的各个进程所接受的选项。100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com