

GCC内联汇编基础

来源：中国自学编程网 发布日期：2008-09-08

这篇文章阐述内联汇编的使用方法。显然，阅读这篇文章您需要具备X86汇编语言和C语言的基础知识。

Contents

- 1. 简介 3
- 2. 概要 3
- 3. GCC汇编格式。 3
 - 1) 源操作数和目的操作数的方向 3
 - 2) 寄存器命名 4
 - 3) 立即数 4
 - 4) 操作数大小 4
 - 5) 内存操作数 4
- 4. 基本形式的内联汇编 4
- 5. 扩展形式的内联汇编 5
 - 5.1 汇编模板 6
 - 5.2 操作数 6
 - 5.3 Clobber List 7
 - 5.4 Volatile...? 8
- 6. 深入constraints。 8
 - 6.1 常用constraints 8
 - 6.2 constraint修改标记 10
- 7. 常用技巧 10
- 8. 结束语 13
- 9. 参考文献 13

1. 简介

[主要是版权/反馈/勘误/感谢等信息。没有翻译。--译者注, 本文中方括号中的都是译者注]

2. 概要

我们现在学习GCC内联汇编，那么内联汇编到底是什么？

[我们首先先来看看内联函数有什么好处]

我们可以让编译器将函数代码插入到调用者代码中，指出函数在代码中具体什么位置被执行。这种函数就是内联函数。内联函数似乎很像一个宏？的确，他们之间有很多相似之处。

那么内联函数到底有什么好处呢？

内联函数降低了函数调用的开销。[不仅仅节省堆栈] 如果某些函数调用的实参相同，那么返回值一定是相同的，这就可能给编译器留下了简化的空间。因为返回值相同了就不必把内联函数的代码插入到调用者的代码中[直接用这个返回值替换就好了]。这样可以减少代码量，视不同的情况而定。声明一个函数是内联函数，使用关键字 `inline`。

现在我们回到内联汇编上来。内联汇编就是一些汇编语句写成的内联函数。它方便，快速，对系统编程非常有用。我们主要目标是研究GCC内联函数的基础格式和使用方法。声明一个内联汇编函数，我们使用关键字 `asm`。