

目标文件格式。不过，无论是什么样的操作系统、ISA 或者目标文件格式，基本的链接概念是通用的，认识到这一点是很重要的。细节可能不尽相同，但是概念是相同的。

7.1 编译器驱动程序

考虑图 7-1 中的 C 语言程序。它将作为贯穿本章的一个小的运行示例，帮助我们说明关于链接是如何工作的一些重要知识点。

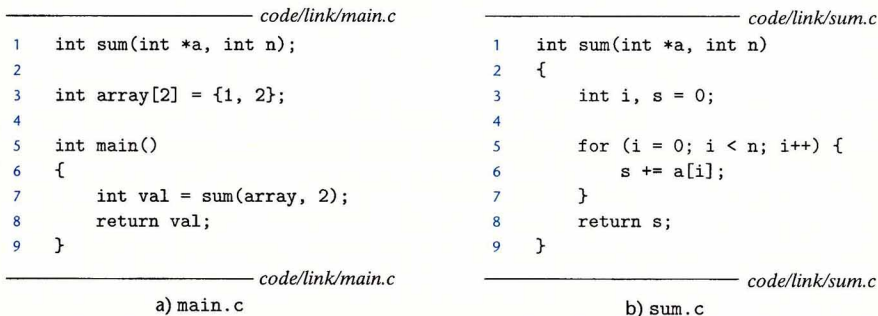


图 7-1 示例程序 1。这个示例程序由两个源文件组成，main.c 和 sum.c。main 函数初始化一个整数数组，然后调用 sum 函数来对数组元素求和

大多数编译系统提供编译器驱动程序(compiler driver)，它代表用户需要在需要时调用语言预处理、编译器、汇编器和链接器。比如，要用 GNU 编译系统构造示例程序，我们就要通过在 shell 中输入下列命令来调用 GCC 驱动程序：

```
linux> gcc -Og -o prog main.c sum.c
```

图 7-2 概括了驱动程序在将示例程序从 ASCII 码源文件翻译成可执行目标文件时的行为。(如果你想看看这些步骤，用 -v 选项来运行 GCC。)驱动程序首先运行 C 预处理器(cpp)[⊖]，它将 C 的源程序 main.c 编译成一个 ASCII 码的中间文件 main.i；

```
cpp [other arguments] main.c /tmp/main.i
```

接下来，驱动程序运行 C 编译器(cc1)，它将 main.i 编译成一个 ASCII 汇编语言文件 main.s：

```
cc1 /tmp/main.i -Og [other arguments] -o /tmp/main.s
```

然后，驱动程序运行汇编器(as)，它将 main.s 编译成一个可重定位目标文件(relocatable object file)main.o：

```
as [other arguments] -o /tmp/main.o /tmp/main.s
```

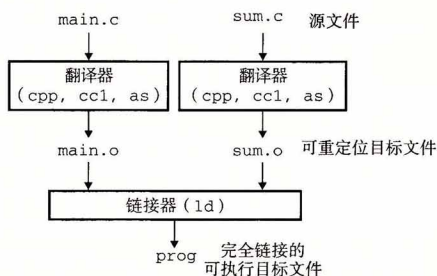


图 7-2 静态链接。链接器将可重定位目标文件组合起来，形成一个可执行目标文件 prog

⊖ 在某些 GCC 版本中，预处理器被集成到编译器驱动程序中。