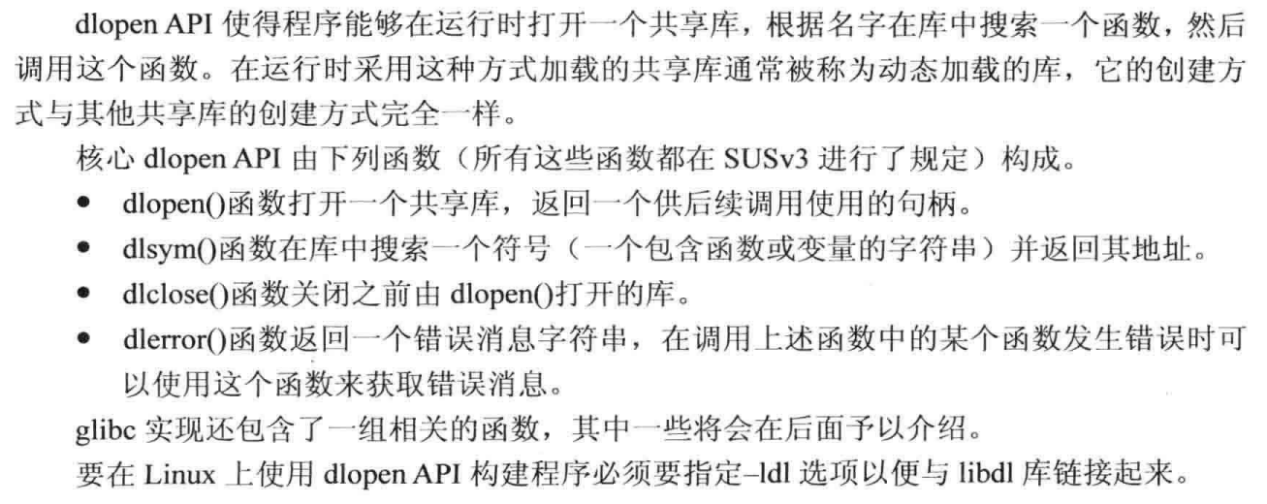
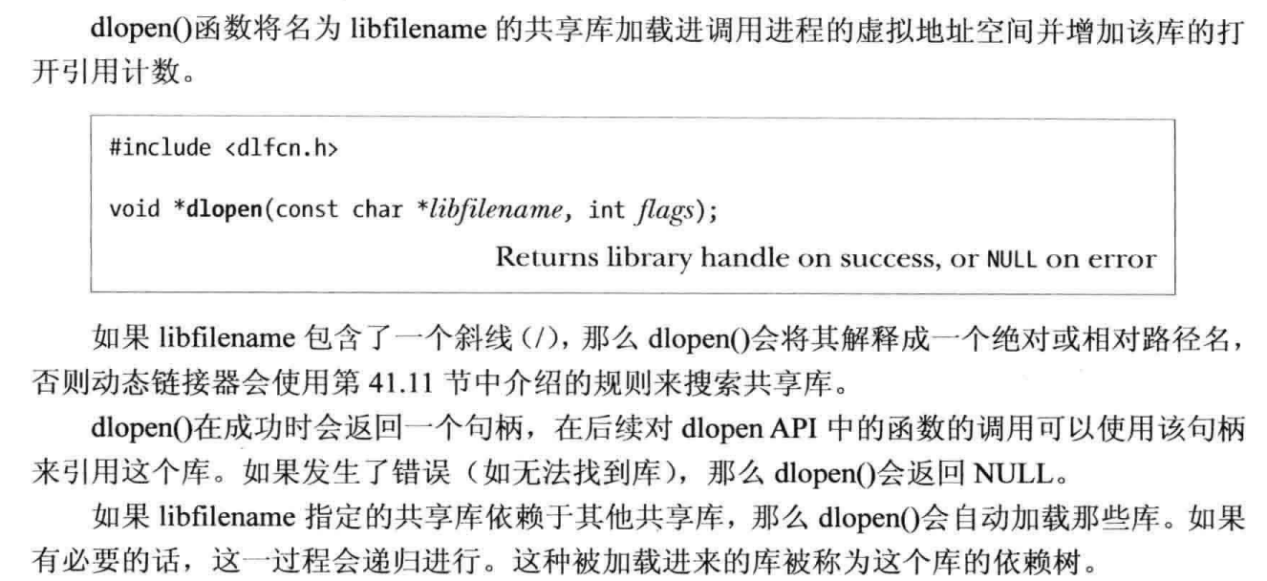
第四十二章

1. 共享库高级特性：
2. 动态加载库：

延迟加载通过dlopen API实现。



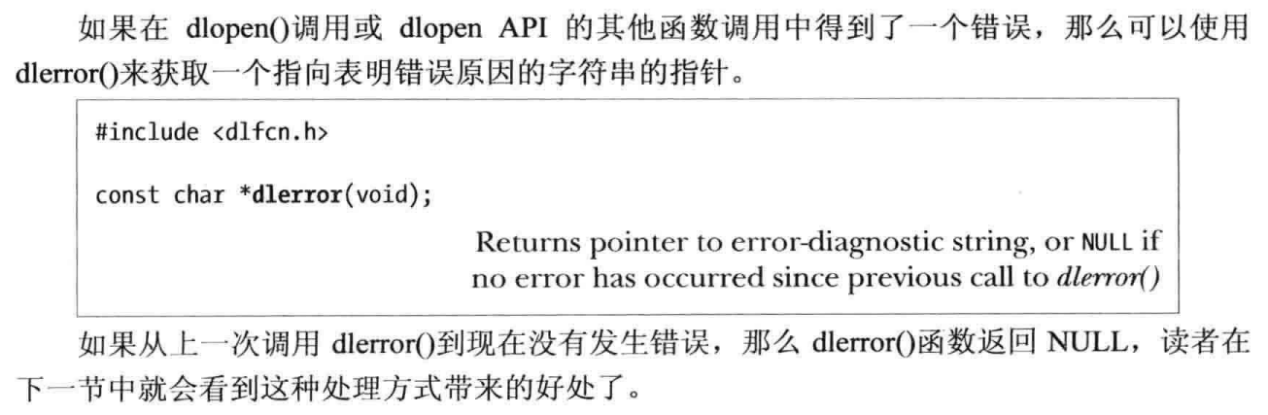
打开共享库：



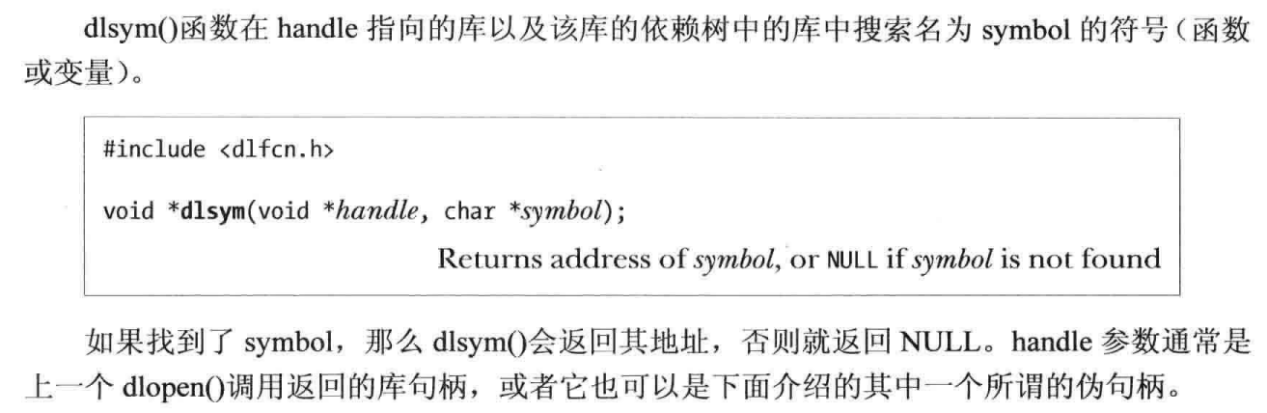
注意，flags里面TRLD\_LAZY和RTLD\_NOW的区别是，前者等到使用该符号的时候才会报错（如果没有的话），后者立马报错。

至于其它flags，查看手册和书本。

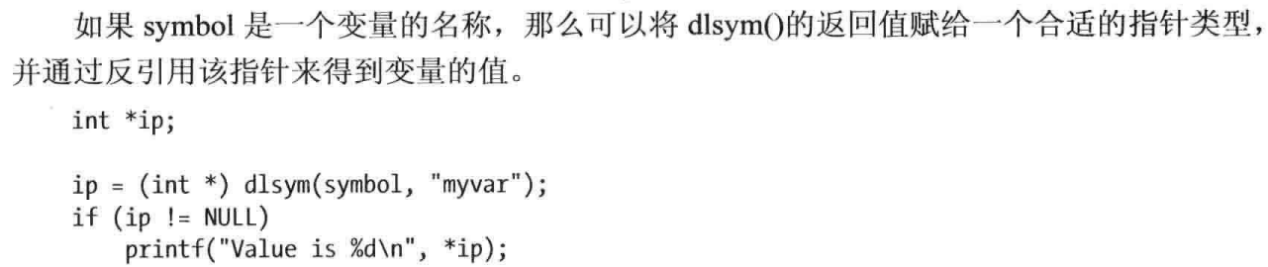
错误诊断：



获取符号地址：



注意，如果符号是变量，获取方式如下：

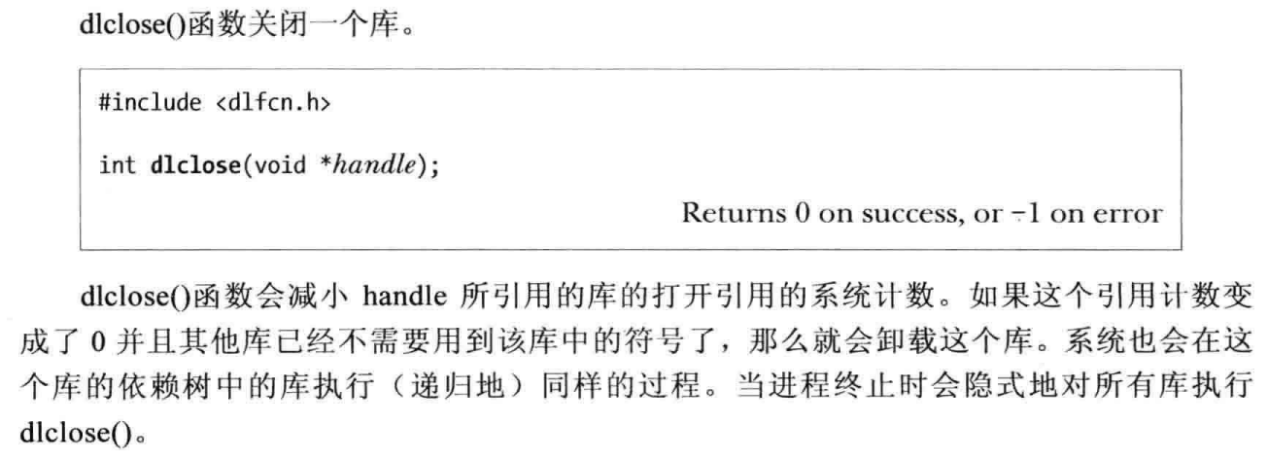


如果是函数：会使用比较笨拙一点的方法：

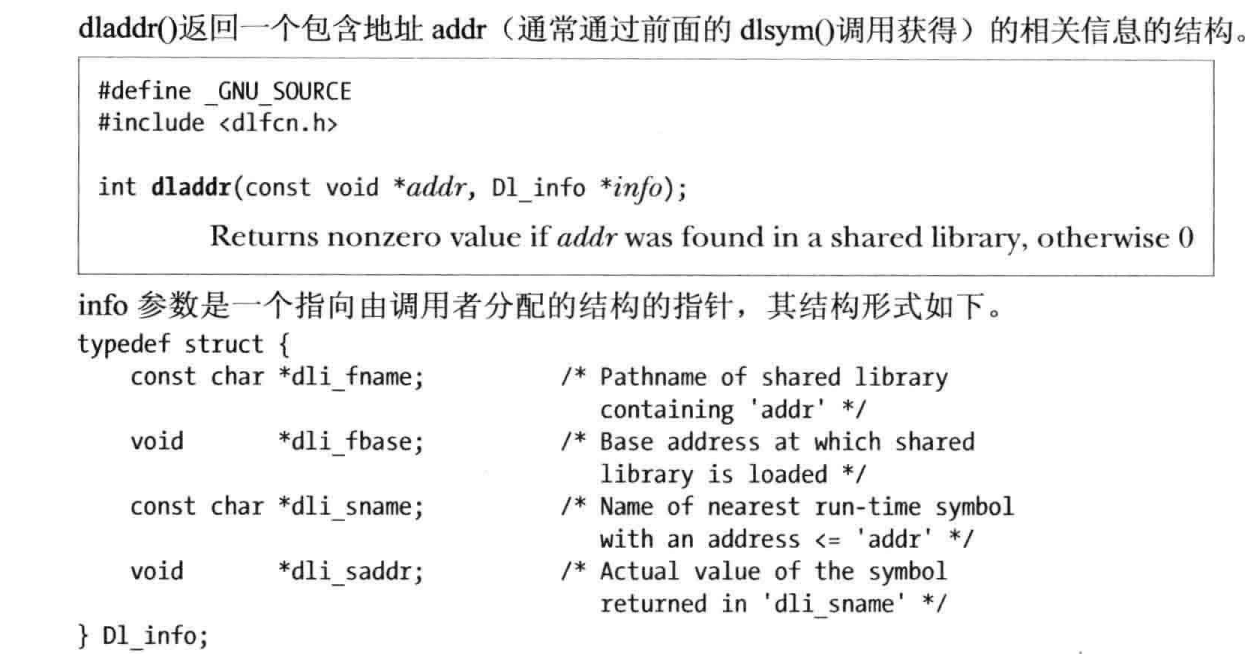


dlsym也可以使用库伪句柄，具体查看手册和书本。

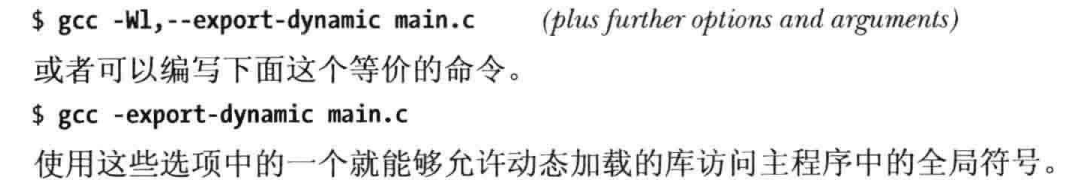
关闭共享库：



获取与加载符号相关的信息：



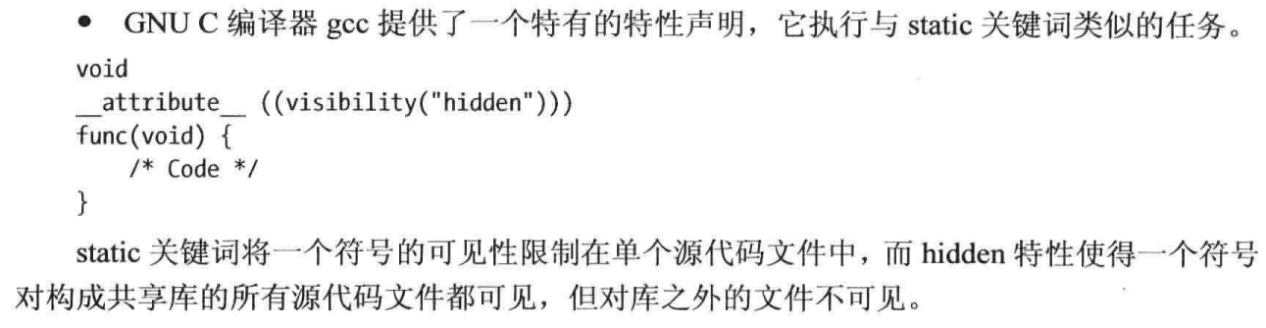
在主程序中访问符号：



1. 控制符号的可见性：

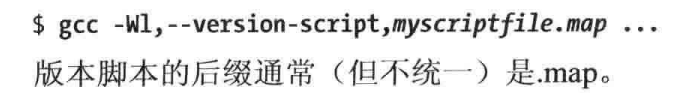
设计良好的共享库应该只公开那些构成其声明的应用程序二进制接口的符号。有什么办法可以控制符号的导出：

1. 使用static
2. hidden特性：



1. 链接器版本脚本：

版本脚本是一个包含链接器ld执行的指令的文本文件。用法：



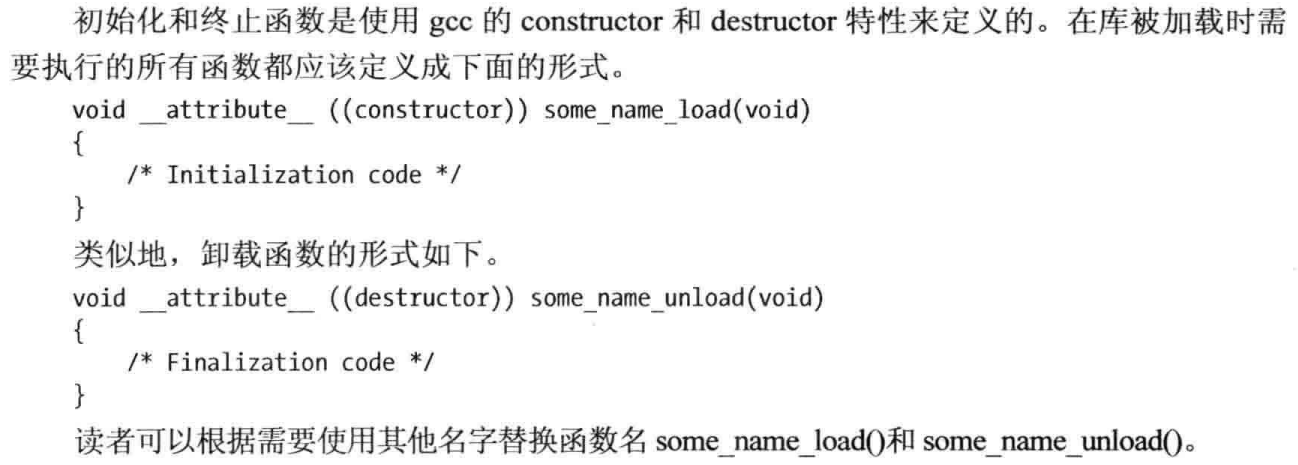
版本脚本的几个用途：

1. 控制符号的可见性：
2. 符号版本号：

具体查看书本。

1. 初始化和终止函数：

可以定义一个或多个在共享库被加载和卸载时自动执行的函数：



1. 预加载共享库：

出于测试的目的，覆盖正常情况下动态链接器寻找符号的规则。

可以使用环境变量LD\_PRELOAD和/etc/ld.so.preload文件。不过后者没有试过。

1. 监控动态链接器：

可以通过LD\_DEBUG环境变量监控动态链接器的操作以弄清楚它在搜索哪些库。LD\_DEBUG\_OUTPUT可以重定向输出到别的地方。