

## 第07章：链接

欢迎来到灰色地带

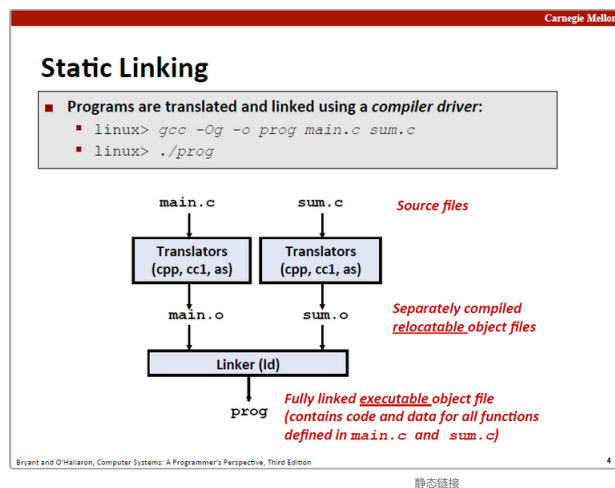
### 视频解说

### 导读

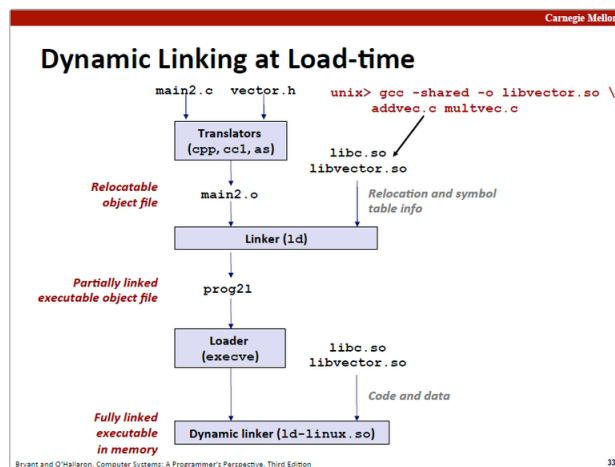
链接（以及装载）处于灰色地带，绝大部分人对它都是懵懂而已，但它们都是非常重要的系统软件。

### 学习方式

CMU教授的视频教程 - Lecture13: 链接



静态链接



动态链接

这里推荐两本书：左边这本是比较经典的：[链接与装载（英文版）](#)，初学者可能会觉得晦涩难懂。

右边这本书：[程序员的自我修养——链接、装载与库](#)，主要介绍系统软件的运行机制和原理，涉及在Windows和Linux两个系统平台上，一个应用程序在编译、链接和运行时刻所发生的各种事项，包括：代码指令是如何保存的，库文件如何与应用程序代码静态链接，应用程序如何被装载到内存中并开始运行，动态链接如何实现，C/C++运行库的工作原理，以及操作系统提供的系统服务是如何被调用的。每个技术专题都配备了大量图、表和代码实例，力求将复杂的机制以简洁的形式表达出来，我认为算是一部国产佳品。



### 重点解读

# Linker and Loader

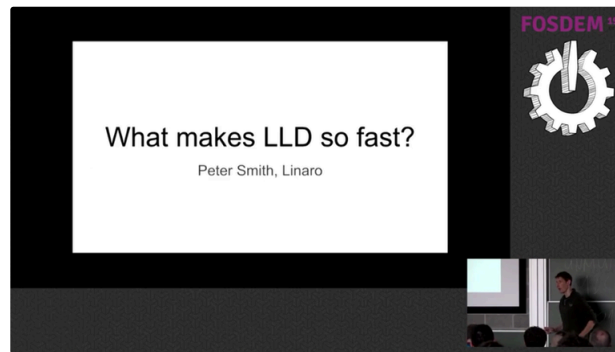
Bin Yang - Senoir Android Framework Architect at Intel Corporation (from slideshare)

## 重点示例

**LD\_PRELOAD**是Linux系统的一个环境变量，它可以影响程序的运行时的链接（Runtime linker），它允许你定义在程序运行前优先加载的动态链接库，一方面，我们可以以此功能来使用自己的或是更好的函数（比如，你可以使用Google开发的tcmalloc来提升效率），而另一方面，我们也可以向别人的程序注入程序，从而达到特定的目的。

```
rock@rock-virtual-machine:~/CSAPP3e/Demo_Chp7$ ls
myfopen.c  myfopen.so  preload_test.c  test test.txt
rock@rock-virtual-machine:~/CSAPP3e/Demo_Chp7$ ./test
Calling the fopen() function...
fopen() succeeded
rock@rock-virtual-machine:~/CSAPP3e/Demo_Chp7$ LD_PRELOAD=./myfopen.so ./test
Calling the fopen() function...
Always failing fopen
fopen() returned NULL
```

链接器：GNU ld vs Google gold vs LLVM lld : <https://gcc.gnu.org/onlinedocs/gcc/Link-Options.html>



<https://av.tib.eu/media/44657>

## 延伸阅读

- 本人拙作（视频讲座）：[Linux环境下：程序的链接、装载和库](#)（比较详细，探讨了很多细节）

Previous	
第06章：存储器层次结构	
Next	
第08章：异常控制流	

Last updated 15 minutes ago