How kernel,compiler,and C library work together

内核、编译器、C语言库是如何在一起工作的？

Kernel 内核

内核是操作系统的核心。在传统的设计中，内核负责内存管理，I/O，处理中断，还有各种各样的其它事情。尽管现在的设计，比如微内核、外内核等等，它把一些服务移植到到用户空间，但对于本文来说无关紧要。

内核通过一组system calls对外提供服务，它们如何被调用，它们做什么取决与不同的内核。

C Library C语言库

首先，当你开始编写你的内核时，你并没有一个可用的C语言库，你必须自己为自己提供这一切，除了编译器本身提供极少的几个部分外，你还必须移植现有的C语言库或者自己重新编写。

C语言库实现了标准的C函数（比如，在<stdlib.h>、<math.h>、<stdio.h>等中声明的事物），并且以二进制的形式提供它们，以便能够让用户空间的程序进行链接。