**第一章的大标题叫做《温故而知新》，顾名思义作者是想让我们先回忆起自己曾经掌握的或者掌握不充分的编程知识，并且引入作者自己的观点。我的想法是正确的，作者在第一节通过HelloWorld这个所有程序员的第一个程序引出了一系列在编译原理方面的问题。在我写代码的时候真的从来没有想过这些问题，更说不上能清楚的知道这些问题的答案。跟随着作者的思路，首先让我通过文章回忆起了计算机的组成，最为核心的部件就是CPU，内存与IO芯片，以及迅猛的计算机硬件结构发展史。作者在谈到系统软件的体系架构中，谈到了重要的协议与接口，并且说明了在软件架构中最重要的层式架构，也就是说每一个中间层都对他下面的层进行包装和扩展。硬件软件的介绍基本完成后该谈到操作系统部分，操作系统最主要的功能就是管理硬件的资源，其实简单来讲就是最初的CPU只能执行单个进程，不支持多线程操作，而现在CPU采用分时调度的模式，所有的应用程序以进程的方式运行在比操作系统权限更低的级别，他们并不是平均分配CPU的执行权，而是采用抢占式的方式获取CPU执行权，最终诞生了现代的操作系统的进程运行方式。接下来作者在设备驱动部分提到，对于操作系统上面的运行库要有一个统一的硬件访问方式，慢慢的成熟的操作系统出现以后程序员就从繁琐的硬件细节中解脱出来，这些事情由操作系统的硬件驱动来完成。我对书中所写文件的读取这部分一知半解，只能大概总结出：文件按照一定的方式以特定的存储空间(字节)存储在扇区(硬盘基本存储单位中)，文件系统内部的运作方式保存了文件的存储结构，也维护了这些数据结构并且保证磁盘中的存储空间能够被有效的利用。循序渐进的，我脑海中闪过一个问题，文件系统不停的处理每一个进程的数据，并且把它们放在硬盘的存储空间中，那么内存不够要怎么办呢？紧接着下一节作者就对这个问题做出了详细的解释。计算机可以通过一种虚拟的内存地址，他不是真实存在的，再通过映射的方法转换成实际的物理地址，这样的话只需要存储虚拟地址到物理地址的映射过程，就能保证地址空间的隔离。谈到分段和分页，在我看来要先清楚为什么需要分段和分页——进程的存在。如果我的系统里只有一个进程，也就是一个正在运行的程序，那么我喜欢访问哪个地址的内存，直接给出任意内存的地址就可以了，也不会影响到任何应用程序，因为整个系统只有一个进程在运行。但是，当有很多进程在跑的时候，我就不能随意的访问任意地方的内存了，在这种情况下，一个进程本身加载到内存中的代码和用到的数据都得不到任何安全的保护，这就是分段和分页的作用。然后作者谈到了线程的概念，在我的理解中，我可以把一个程序例如杀毒软件看作是一个进程，而杀毒软件上的‘全盘扫描’，‘垃圾清理’，‘空间优化’等等内容可以看作是一个线程，也就是轻量级进程，然而不管是多处理器还是单处理器的计算机，总是会出现多线程并发执行的，为了合理的分配CPU的执行权，在线程调度模式中采用优先级调度和轮转法来保证多个线程的并发执行。Windows系统与linux系统中现成的优先级不仅可以手动调正，还可以根据实际的需要自行调整以保证调度的高效率执行。更重要的是，多线程并发执行总是会随时随地带来一些线程安全问题，也就是多线程程序再并发时数据的一致性的安全问题，经典的线程安全问题的解决方案有同步与锁，在读者给出的程序实例当中我也了解到使用同步方法与锁方法解决线程安全问题的不同之处，对于在我以后更深入的编写多线程高并发相关程序有着很大的帮助。总的来说，第一章真正做到了帮助我在计算机知识上实现了——温故而知新。**

**第六章——可执行文件的装载与进程。在这一章中我充分地了解到ELF文件在linux下的装载过程，耕种套的是可执行文件装载的本质到底是什么。在第一节中作者首先详细的介绍了进程虚拟地址空间，让我了解到每个程序在运行之后都有自己独立的虚拟地址空间，这是一个非常抽象的概念，虚拟地址空间的大小也由操作系统决定，32位的操作系统虚拟地址空间的大小就是4G，可是实际是用不到4G的，用到的内存会通过页表映射到物理内存，所以每个进程都可以使用同样的虚拟内存地址而不冲突。并且在32位的CPU下，如果时虚拟地址空间，那就不可以使用超过4GB的内存，因为指针的限制，如果是实际的内存空间是可以的。接着，作者在书中谈到了装载，简单理解，程序需要的所有指令和数据必须在内存中运行才能保证程序的正常运行，那么最简单的方式就是把这一大堆东西都装进内存当中，可是这样是非常浪费资源的，所以通过动态装入我们可以在不添加内存的情况下让更多的程序正常运行起来，覆盖装入和页映射就是两种非常好的动态装载方法，也就是说程序用到什么，他就把这些东西装入内存，如果用不到就不妨到内存中，而是存放在磁盘当中。值得一记的是，覆盖装入是一种折中的方案，覆盖装入是典型的利用时间换取空间的做法，覆盖装入在发明虚拟存储之前使用比较广泛，现在基本被淘汰了。而页映射属于虚拟存储机制的一部分，利用程序的局部性原理，不会一下子就把程序所需要的所有东西装进内存，而是按页进行划分的。接下来作者又带领我从操作系统的角度看待可执行文件的装载，让我充分地了解到了进程的建立过程，通过之前提到的虚拟地址空间，将可执行文件便造成一种虚拟的映像文件，这样将CPU指令寄存器设置成可执行文件入口，就可以启动进程了。**

**这两章的内容既帮助我回忆起了以前学习过的知识，加深了我对旧知识的印象，同时还能从中深挖出各种各样的问题，更丰富了我对这些计算机知识的理解；然而，对于我这种水平不高的编程学习者，书中难免有些内容是有点深刻难懂的，例如第六章的ELF文件以及Linux部分我看了很多遍还是迟迟难以理解。但我觉得没关系，毕竟我从这本书中还是学习到了不少非常有价值的东西，并且这本书还增加了我对编程的兴趣，激发我想更加深刻的了解编程世界的动力。这本书启发我在以后的学习过程中，多了解一些事物的底层的基础的东西，多研究事物的原理以及过程，不仅是编程而是在方方面面都要保持清醒深刻的头脑，希望这本书能伴随我的编程路途，时不时都回来看看，“温故而知新”一下。**