第一章

通过阅读第一章，我对计算机软硬件体系里几个比较重要的概念有了一定的了解（以前没有主动地了解这方面的知识）。

首先是我对CPU与外围部件的连接方式的简单理解。早期与现在的计算机好像都是通过总线连接在一起。早期计算机CPU的核心频率与内存的频率一致，所以可以直接连接在同一个总线上，I/O设备则因速度较慢而只能通过响应的I/O控制器才能与总线相接。而后来，因为CPU核心频率的提升，内存与CPU无法使用同一个总线直接相连，也就需要工具（比如芯片）来帮助CPU与系统总线的建立通信。

然后是多处理器（SMP）与多核处理器。根据书本，两者除了在缓存共享等方面有细微的差别，基本可以看作同一个概念。我对多处理器的理解是：因为CPU的频率目前已发展到现阶段的上限，为了继续提高处理速度，就只能增加CPU的数量。但因为每个CPU的地位和功能是基本相同的（类似于并行，它们也是相互独立的），所以SMP比较适合处理大量且相互独立的请求。

最后是我对第一章提及的部分概念的理解：

操作系统有点类似于网络管家，既能提供抽象的接口，又能管理硬件资源。

地址空间分为两种：虚拟地址空间和物理地址空间。物理地址空间是实实在在存在于计算机中，而且对于每一台计算机来说只有唯一的一个，与计算机地址线、内存以及外部I/O设备有关。虚拟地址空间是指虚拟的、人们想象出来的地址空间，实际并不存在，每个进程都有自己独立的虚拟空间，而且每个进程只能访问自己的地址空间。而想要实现虚拟地址到物理地址的转换（实现虚拟存储），则需要MMU来完成页映射。

线程是程序执行流的最小单元。一个标准的线程由线程ID、当前指令指针（PC）、寄存器集合和堆栈组成。在阅读时，我对书上线程与进程的关系图不太理解。通过查阅资料，以下是我的一点理解。进程是CPU所能处理的单个任务，而线程则有点像其中一个或多个个体，共享着进程的空间，共同协作完成这个任务（进程）。

第六章

第六章主要内容为介绍可执行文件的装载与进程。

程序是静态的，进程是动态的。进程是程序运行时的一个过程，而程序是预先编译好的指令和数据的集合。每个程序被运行起来以后，都将拥有自己独立的虚拟地址空间。虚拟地址空间的大小由CPU的位数决定的。而硬件则决定了地址空间的最大理论上限。

程序执行时，指令和数据必须在内存中才能够正常运行。将指令和数据装入内存的方法分为静态装入和动态装入两种。静态装入是直接将程序运行所需要的指令和数据全都装入内存中，而动态装入则是将程序最常用的部分驻留在内存中，而将一些不太常用的数据存放在计算机磁盘里。动态装载的两种典型的方法是覆盖装入和页映射。覆盖装入是各模块轮流共用存一片区域，程序员手动控制哪个模块在某一时刻进入该区域。页映射是将内存和所有磁盘中的数据和指令按照“页”为单位划分成若干个页，以后所有的装载和操作的单位就是页。

在有虚拟存储的情况下，上述过程最开始只需要做三件事情：创建一个独立的虚拟地址

空间。读取可执行文件头，并且建立虚拟空间与可执行文件的映射关系。将CPU的指令寄存器设置成可执行文件的入口地址，启动运行。

最后两节深入介绍了Linux和Windows系统下静态链接的ELF和PE可执行文件的装载和运行。我个人因为之前没有接触过相关的知识，所以这部分内容对我而言比较难以理解，无法总结出比较关键的点。我会在后续的学习中慢慢消化这部分知识点。