总结

第一章

介绍了计算机运行时的各个硬件发生的流程，计算机有三个最为关键的部件，为CPU、I/O控制芯片和内存，介绍了CPU从提高频率，到后来因为技术到达上限开始发展多线程发展。提出了解决任何计算机问题都可以通过增加一个中间层来解决。讲述了CPU在运行过程中，为了提高运算能力，能够同时处理多项任务，然CPU以多任务系统处理任务，防止计算机因为一个线程无法运作导致整体瘫痪。随着计算机处理能力的提升，硬盘随之发展，文件通过分区域储存在硬盘中，通过IDE接口由CPU进行读取。在计算机进行内存分配时，会产生地址空间不隔离、内存使用率低、程序运行时地址不确定的问题，人们通过以虚拟地址映射现实的物理地址来解决第一个和第三个问题，及把虚拟地址映射到物理地址的每一段上，因此当计算机访问空间超出一定范围就会判断为一个非法访问（segmentation fault）。为了解决内存使用率低的问题，采取了分页的方式，把程序中的各个部分分为多个虚拟页，把有用的部分映射到物理页，没用的部分存放在磁盘页，当需要访问磁盘页中的内容时，计算机发出page fault使系统把磁盘页中的虚拟页映射到物理页中进行访问。最后介绍了系统的线程，及系统进行一个程序的过程。

第六章

以32位系统为例，有4G的内存，其中可能系统本身要占用一个G的内存，而用户使用的仅有3G的内存，在这3G中，用户真正能使用到的往往只有2G左右，为了解决系统内存不足的问题，Windows系统有PAE（通过修改映射的方式，使得可访问更多的物理内存）和AWE（在物理空间形成多个窗口，将虚拟空间需要使用的窗口映射到不同的物理空间）的方法来补救32为地址空间不够的情况。为了解决程序浪费内存的情况，有覆盖装入和页映射的方法。覆盖装入及程序员用一个辅助程序把各个形成树状结构的模块，把根部的模块放到上一级的空间中使用。而在页映射中，整体以第一章相同。系统创造一个进程，装载可执行文件并执行的过程大致分为以下三步，创造一个独立的虚拟空间地址，读取可执行文件头并建立映射关系，将CPU的指令寄存器设置在可执行文件入口地址并执行。在第二步中，可执行文件头被放在虚拟段（VMA），CPU读取到虚拟段产生的空白页，发生页错误，把操作交给操作系统，操作系统计算文件大小，并映射到物理地址。因为系统只关心装载问题，为了节约系统内存空间，系统把相同权限的section放在同一个segment中。操作系统通过给进程空间划分一个个VMA来管理进程的虚拟空间，基本原则是将相同权限属性有相同映像文件的映射分成一个个VMA。