**《程序员的自我修养》读后总结**

**第一章：**

首先我加深了对计算机的理解，比如在计算机中最重要的部件有三个，分别是中央处理器CPU、内存和I/O控制芯片，然后了解了由于对性能要求不断地提升，cpu的频率不断提高直到物理极限。性能上来了，还有什么方面能提高呢？利用率。于是，多道程序以及随后的分层系统应运而生，操作系统初具雏形。直到现在的抢占式，，操作系统可以强制剥夺CPU资源并且分配给它认为目前最需要的进程。

接着书中就带领我领略文件系统：硬盘以扇区为基本存储单位，将字节存储在扇区中，需要读取这些字节时，就用到read系统。

接下来深入到内存的分配的问题，通过书中的介绍，我第一次知道内存空间需要相互隔离避免被破坏，而且物理地址效率低下，程序运行不稳定。而怎么解决呢？从中引入了一个中间层的概念，通过间接访问虚拟地址再映射到物理地址。

分段分页：对于这两个方法给我的感觉就是抽象，从书中阅读，我认为虚拟地址映射到物理地址，由于虚拟地址映射后物理地址的不同从而起到了隔离的作用，但对于程序运行的地址是如何确定的我始终都有种模糊和朦胧。而分页就是不同虚拟地址空进程的虚拟地址空间按页分割，常用的数据和代码页装载放到内存中，不常用的代码和数据放在磁盘里，当需要用到的时候再把它从磁盘里取出来。虚拟空间的页叫虚拟页，物理内存中的页叫做物理页，磁盘中的页叫做磁盘页。

**第六章：**

程序执行时所需要的指令和数据必须在内存中才能够正常运行，最简单的办法就是将程序运行所需要的指令和数据全都装入内存中，这样程序就可以顺利运行，这就是最简单的静态装入的办法。当所需内存大于物理内存就需要“开源节流”，添加内存（较难）和提高内存利用，而后者就需要通过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磁盘里面，这就是动态装入。

树状结构：由于模块之间具有相互调用依赖关系，并且模块数量多，就衍生出树状结构。在树状结构中， 任何一个模块到根模块都叫调用路径。禁止跨树间调用。任意一个模块不允许跨过树状结构进行调用。

一个进程可分成VMA区域，代码VMA，权限只读、可执行；有映像文件。数据VMA，权限可读写、可执行；有映像文件。堆VMA，权限可读写、可执行；无映像文件，匿名，可向上扩展。栈VMA，权限可读写、不可执行；无映像文件，匿名，可向下扩展。