第二章：

操作系统怎么去**管理物理内存**

计算机体系结构：

三大块

CPU：**完成对整个程序和软件的执行的控制**

内存：放置程序的**代码**和处理的**数据**

外设（I/0）

内存的层次结构：

寄存器、**cache、（操作系统没有办法直接管理）速度很快、容量很小（位于cpu芯片内部）**

、**主存/物理内存（放置操作系统本身，和运行代码）一掉电数据便没了**

硬盘：需要**永久保存**的代码，容量很大

**逻辑地址空间、独立地址空间（隔离地址空间）**

**暂时不需要访问的数据可以放到硬盘上去，需要访问的数据放在内存中**

操作系统中**管理内存**的方法：

**程序重定位：**

**分段：**

**分页：**

**虚拟内存：**

**按需分页虚拟内存：**

地址空间生成：

**物理地址空间、逻辑地址空间**

**物理：直接对应硬件，内存条和硬盘，主存**

**逻辑：运行的程序看到的地址空间**

逻辑的空间怎么生成：

C语言中函数名、变量名就是一个逻辑地址

通过**编译器**将**符号地址变成逻辑地址**，.o文件从0开始，link功能将多个.o合并成.exe.再存入内存中

**Loader应用程序将程序放在内存中运行，此时会给分配内存的地址。**

运行程序放在内存中后还是逻辑地址，运行时对应的地址是物理地址。

Cpu从**内存**中取出来**指令**，知道其**逻辑地址**，**查找对应的物理地址**在什么地方。其对应关系在**MMU中，存储这逻辑地址和物理地址的对应关系的表（逻辑地址映射表）**。Mmu中没有就会到内存中去找。Cpu获取该物理地址的内容，内存将内容通过总线给cpu，cpu根据指令内容开始执行。

操作系统建立一个**逻辑地址到物理地址的映射关系**。

地址安全检查：

操作系统设置了**逻辑地址的基址和界限**（长度）（是一个表）（起始地址和长度）

确保每一个程序的**有效访问的地址空间**。

但逻辑地址不满足区域限制，则会触发**内存访问异常**

连续物理内存分配：

内存的碎片问题：

出现无法进一步利用的空间（碎片）

外碎片和内碎片：外碎片是**分配单元之间的内**存，内碎片已经**分配给应用程序的单元中不能进一步使用的空间**

操作系统什么时候进行连续内存空间分配：

1.**操作系统需要把以应用程序从硬盘加载到内存中去，在内存中分配一块连续的区域让程序运行**

2.**应用程序在运行时需要访问数据，给数据分配空间**

内存分配算法：

首次适配first fit：

**从低地址到高地址找**，**第一个满足分配**需求的空间

按地址排序的空闲块列表，寻找一个合适的空间

当分配一个空闲块之后，会把不需要的部分分出了一个空闲块

回收过程中需要考虑是否需要合并空闲块，让他更大一下

最佳适配best fit：

找到能都满足需求的**最小的空闲块**

**外碎片产生最小化**，**避免对大空闲块拆分**

外碎片拆得很细，**进一步利用的情况比较少**

最差适配Worst fit：

找到**空闲块大小差距最大的**。

把大块变成小块，**避免太多的微小的碎片**

大块的处理，可能会下次分配没有空间分配了

压缩式碎片整理：紧致算法compaction，**重置程序的位置，（内存的重新拷贝过程）合并碎片**，将程序都向上挪动，拷贝，让程序紧致一起，这样空闲块便能合并了。（**程序的重定位，重定位的时间，不能在程序运行的时候，以及开销问题**）

换入换出：swapping 交换式碎片整理

**运行的程序需要更多的内存**，将**等待的程序的空间从主存（内存）放到硬盘**上去，将回收的空间分配给运行的程序。

怎么挑选程序去换入换出