操作系统第五章作业

Created by 黄子曦 23009290041

**2，6，15，16，17**

T2

地址重定位：程序的逻辑地址映射到物理地址的转换过程

分类：静态再定位和动态再定位。

各自特点

  静态再定位：指在程序运行之前就将地址映射过程完成；

     AD：无需硬件支持，容易实现

     DIS：程序定位后不能再移动，因此不能重新分配内存

  动态再定位：是在程序运行的过程中，访问内存之前，再进行转换。

   AD：运行过程中程序可以移动，可以换入换出，因此可以较好利用较小的内存块，不必分配连续的大空间，有利于解决内存紧张问题

   DIS：需要附加的硬件支持，内存管理单元【包括基质寄存器和变址寄存器】

T6

解题反思：两者都需要一次性全部装入【没有虚拟内存技术的情况下】

段号、页号都是从0开始，段内偏移W也是从0开始

**分页存储管理**：每一页的大小一样，进程在内存中不一定连续分配，其本身并无完整的意义，因而不便于实现信息共享

**分段存储管理**：每一段的大小不一样，段是信息的逻辑单位，更有利于实现保护【防止别的程序段对内存中的该程序造成污染】，也更有利于实现动态链接

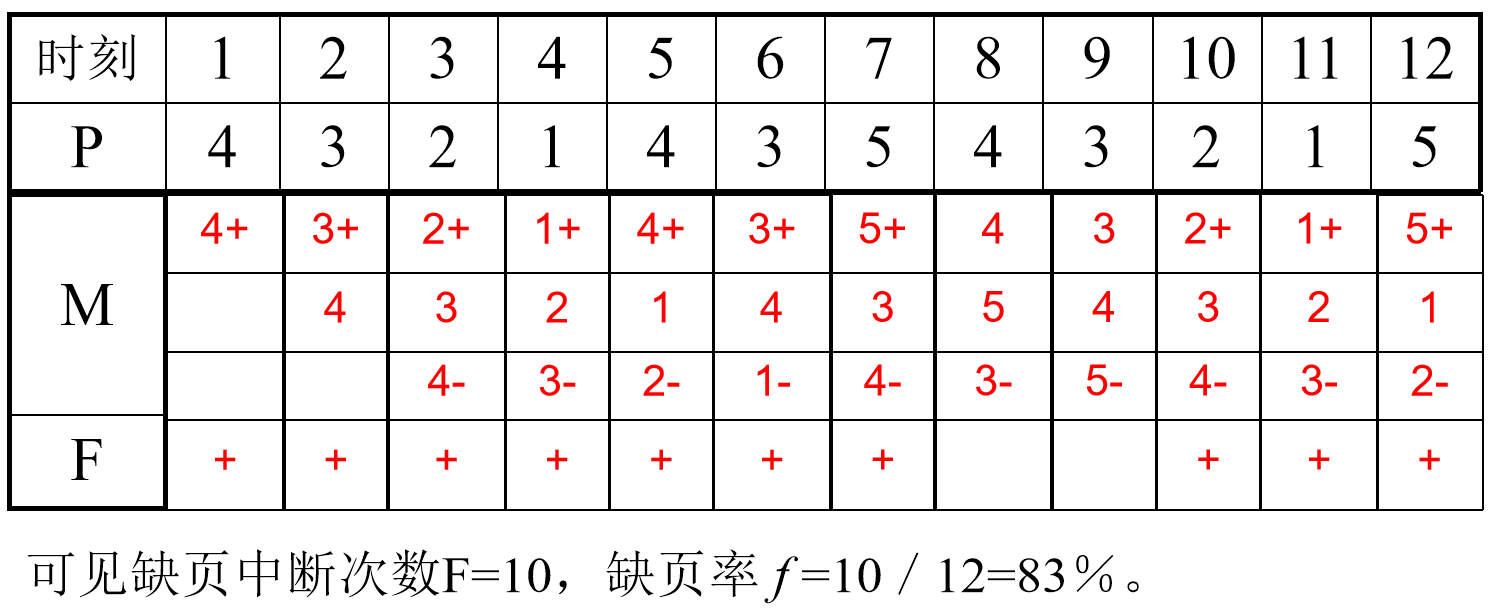
**动态链接：运行到某一段的时候，才将其调入内存，要求以段作为划分单位**

虚页& 实页：

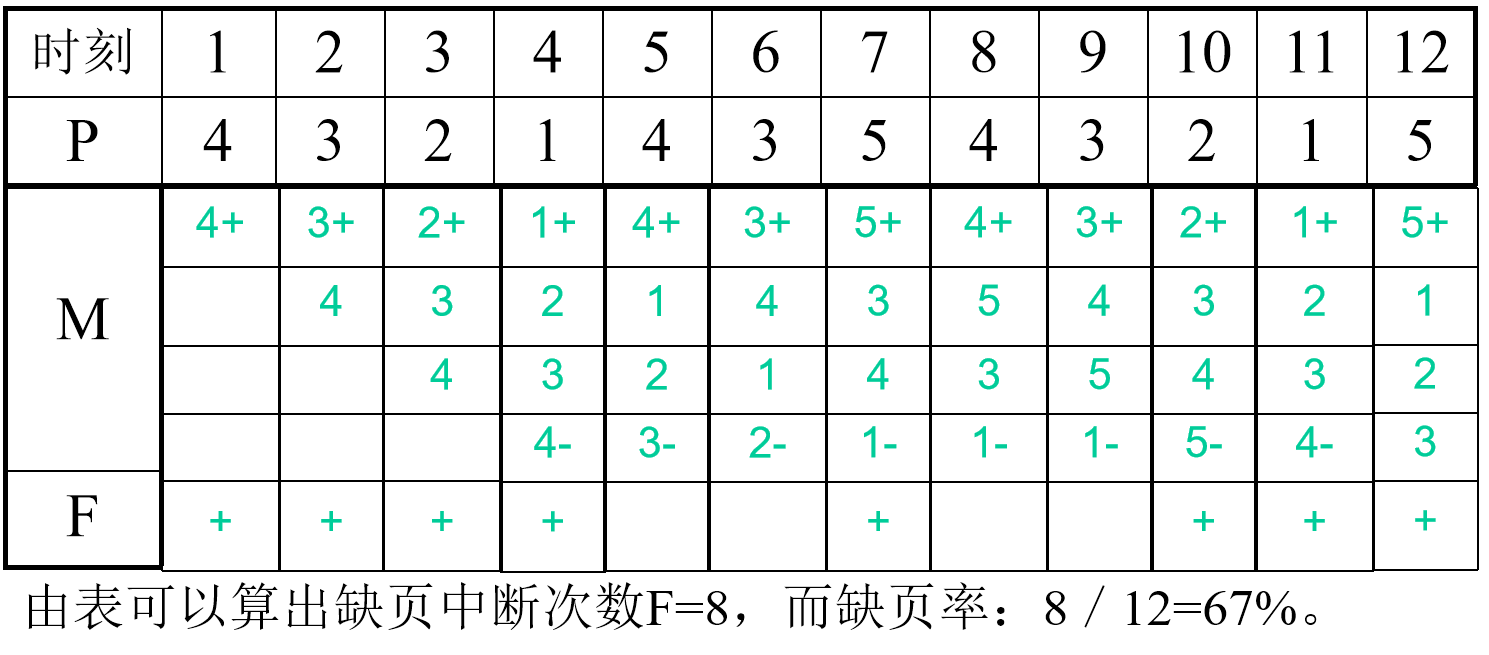
把作业的**虚拟地址空间/**主存划分成若干个长度相等的页（Pages），称为**“虚页”**/**实页**。4

T15

【LRU，M=3】



【LRU，M=4】



启示：在一定范围内，采用同种替换策略，适当增加M的数量，能够显著降低缺页率

T16

解题反思：访问段表控制寄存器，获得段表位置？这是访问寄存器！不是主存！

（1）

1. 检验段号合法性：保证段号小于段表长度
2. 获取段表起始地址，然后在段表的对应段号处，找到真实的物理起始地址
3. 完成地址变换：将上述起始地址和段内偏移量拼在一起，就得到了真实地址

（2）怎么加？直接数值相加！ 219+340 = 559

559；3310；**段内地址超过段长，越界**；590；1637

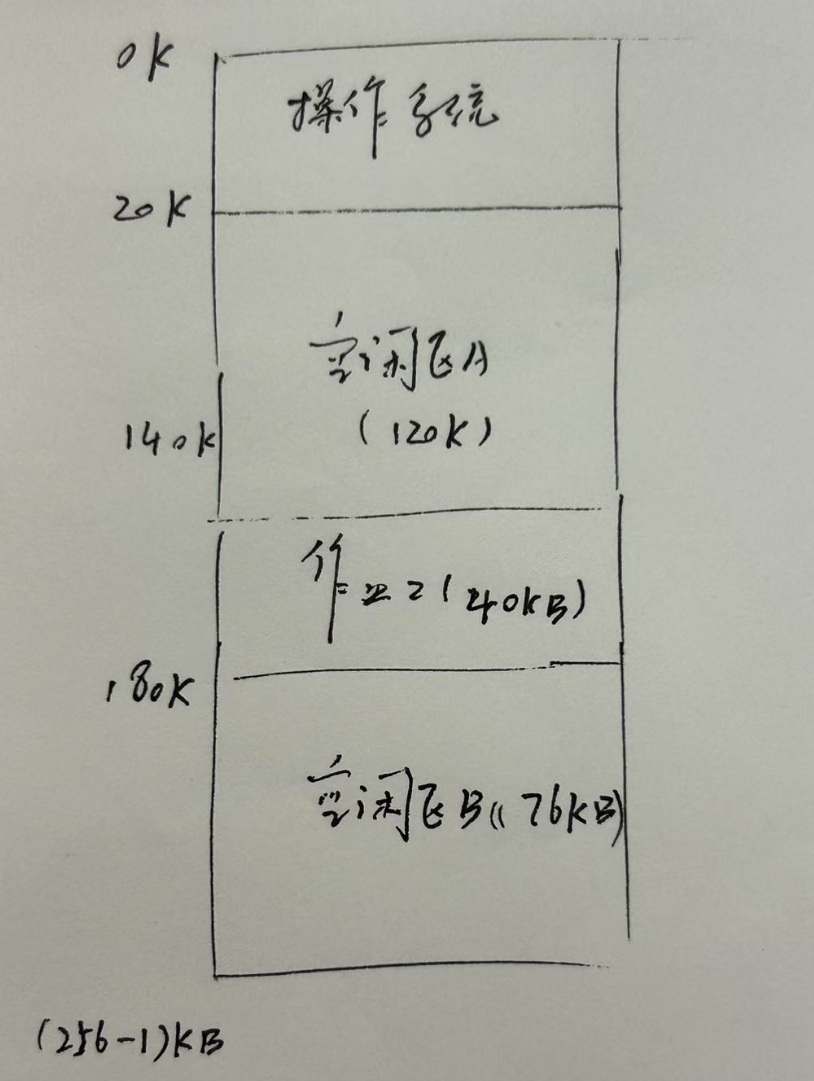
（3）访问段表，完成地址变换后再访问主存，**一共两次**

T17

解题反思：首次适应分配算法和最佳适应分配算法在释放的时候都是尽量合并成更大空间

只不过对空闲空间的排序：一个是按照地址排序，一个按照空闲区大小来排序

1. 作业1，3执行后，释放的空间分别与旁边的空闲区连成更大的空闲区，分别构成了120K和76K的连续空闲区
2. 由两种算法的特征得，二者只在对空闲区块的排序方式上有差异，**主存的结构均为下图所示**



1. 首次适应：按照地址先后查找：70K的作业放在120K的内存中；最佳适应：按照空闲空间大小从小到大查找，最终放在76K空闲区中