

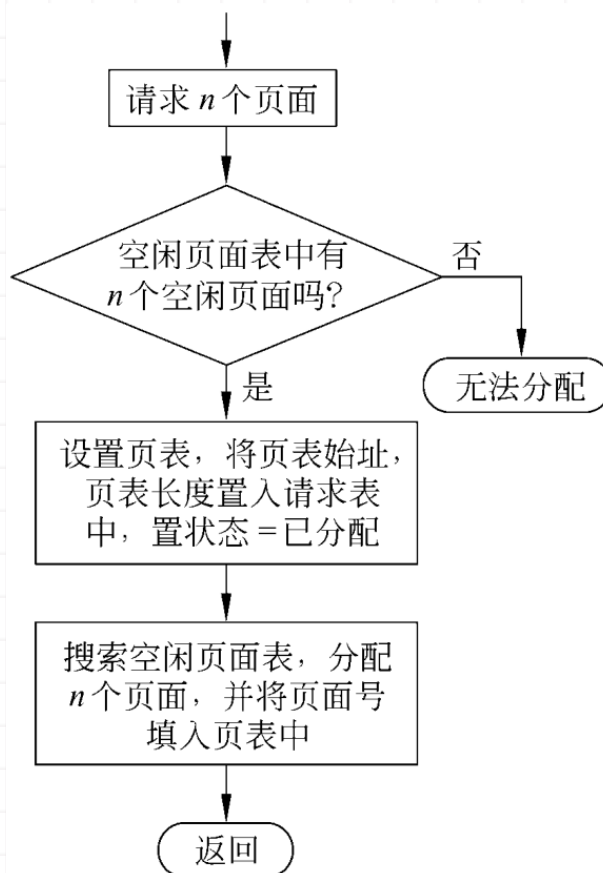
主题1 Part2

6. 设计两个程序实例，分析说明程序运行时的时间局限性和空间局限性，进而说明虚拟存储器的基本工作情况和关键技术。

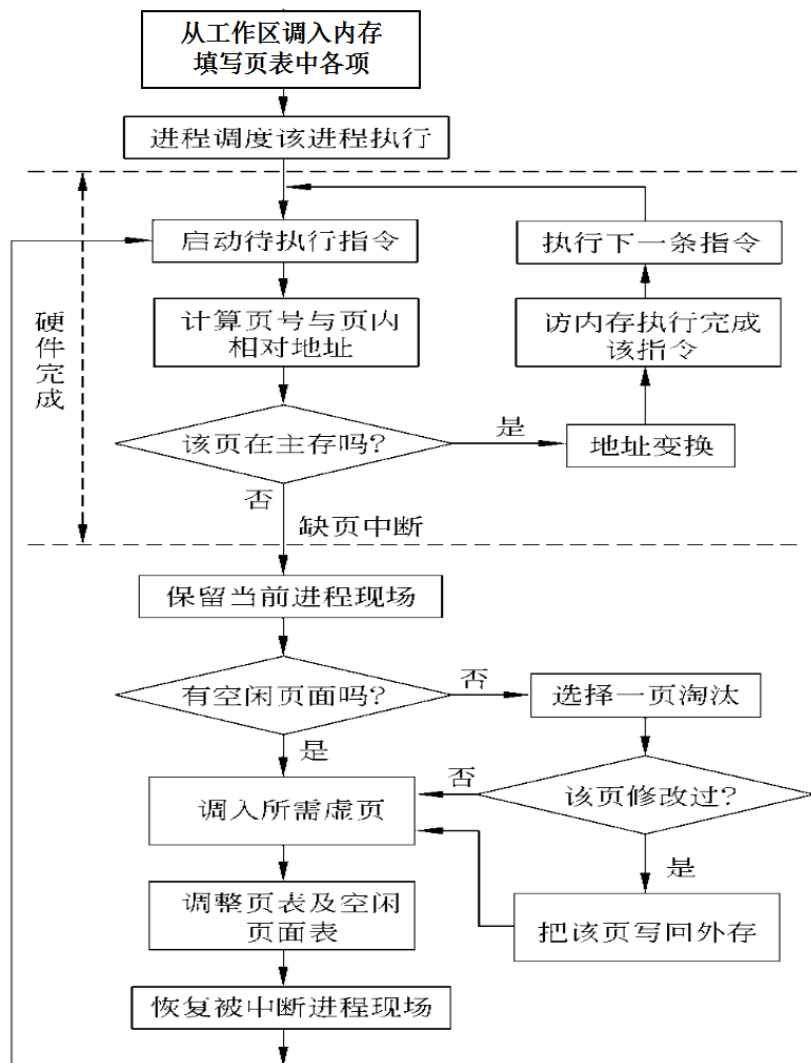
这个题没有小组讲到

7. 以一个程序空间为10页的进程为例，假设在请求页式管理方式下获得4个页框的内存物理空间，请对比分析其基本页式管理和请求调页管理方式下的页表机制的异同，并说明请求调页管理中的各页表项的作用和使用过程。

基本页式管理与请求调页管理是很容易弄混的一对概念，基本页式管理如下图所示：



请求调页管理如下图所示：



其中，二者也与程序的局部性原理有关。程序在执行时呈现出局部性规律，即

- 在一段时间内，整个程序的执行仅限于程序中的某一部分。
- 相应地，执行所访问的存储空间也局限于某个内存区域。

8. 实现虚拟存储器需要哪些硬件支持？设计一个进程实例，说明请求调页管理方式下程序访问数据的过程。可能出现几种不同情况？分析各种情况的系统消耗。要求：设计的实例涵盖各种可能发生的数据访问情况。

第十四组首先分析了虚拟存储器的硬件支持。即以下3点：

1. 请求页表机制
2. 缺页中断机构
3. 地址变换机构

其中，地址变换机构是最需要我们理解与掌握的。在请求页表的过程中，若对应页面未调入内存，则产生缺页中断，之后由操作系统的缺页中断处理程序进行处理。由于快表中的页面一定是在内存中的，所以若某个页面从外存调入内存，快表中的相应表项要进行修改。

十四组还针对了不同情况展开分析，例如不在页表和快表中，且页表和快表未满；不在快表中但在页表中等等情况，需要我们区分清楚。对于系统消耗而言，总体来说：

1. 页在内存或在快表时，系统消耗较小。
2. 页不在内存中或内存满时，系统消耗较大。

9.什么是缺页中断处理程序？它是如何工作的？分析其中CPU状态的变化过程及其特殊性，比较缺页中断机构与一般中断之间的异同。

第四组	以下几个方面介绍缺页中断处理程序： <ol style="list-style-type: none">1. 缺页中断的处理过程2. CPU状态变化3. 缺页中断与一般中断的对比 其中最令人印象深刻的是一般中断与缺页中断的对比。对于一般中断而言，执行完每个指令之后，CPU都要检查是否有外部中断信号如果检测到外部中断信号，则需要保护被中断进程的CPU环境根据中断信号类型转入相应的中断处理程序(进入内核态)在最后恢复原进程的CPU环境并退出中断，返回原进程继续往下执行。而此处的缺页中断其实就相当于其中的一个特殊情况，只不过发送的中断信号是缺页中断信号。
-----	---

10.进程的页面引用串为：4，3，2，1，4，3，5，4，3，2。若系统分配给该进程3块内存物理块。请分析采用最优算法，FIFO，时钟算法和LRU页面置换算法的页面置换过程，计算页面访问的命中次数。

第2组	详细的介绍了常用的页面置换算法： <ol style="list-style-type: none">1. 最佳置换算法2. LRU最近最久未使用3. FIFO页面置换算法4. CLOCK页面置换算法 并从数据结构实现的方式介绍了这部分。这部分也和我们的实验内容、作业相辅相成。其中LRU算法的实现方法十分重要。具体为：赋予每个页面对应的页表项中，用访问字段记录该页面自上次被访问以来所经历的时间t。当需要淘汰一个页面时，选择现有页面中t值最大的，即最近最久未使用的页面。这也对硬件有要求，具体可分为下面几种： <ol style="list-style-type: none">1. 计数器/计时器2. 移位寄存器3. 特殊的栈 在实验中，我使用了第三种方法，即栈实现。
-----	---