

4.2 存储管理

主要考点

- 1、分区存储管理
- 2、分页存储管理
- 3、分段存储管理
- 4、段页式存储管理
- 5、虚拟存储管理
- 6、页面置换算法

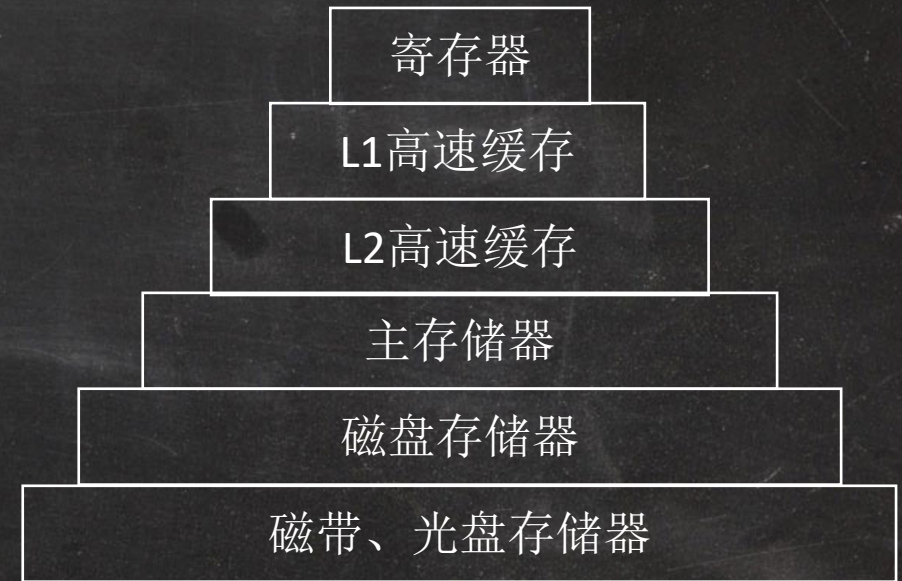
基本概念

- 逻辑地址（虚拟地址、相对地址）：

程序员使用的地址只是用符号命名的一个地址，称为符号名地址，这个地址并不是主存中真实存在的地址。

- 物理地址（绝对地址）：

是主存中真实存在的地址。



图：存储器的层次结构

地址重定位

- 一个程序，没有运行时，存储在外存，程序运行时，需要装载到内存中，就需要把程序中的指令和数据的逻辑地址转换为对应的物理地址，这个转换的过程称为地址重定位。
- 静态重定位：在程序装入主存时已经完成了逻辑地址到物理地址的变化，在程序的执行期间不会再发生变化。
- 动态重定位：在程序运行期间完成逻辑地址到物理地址的变换。

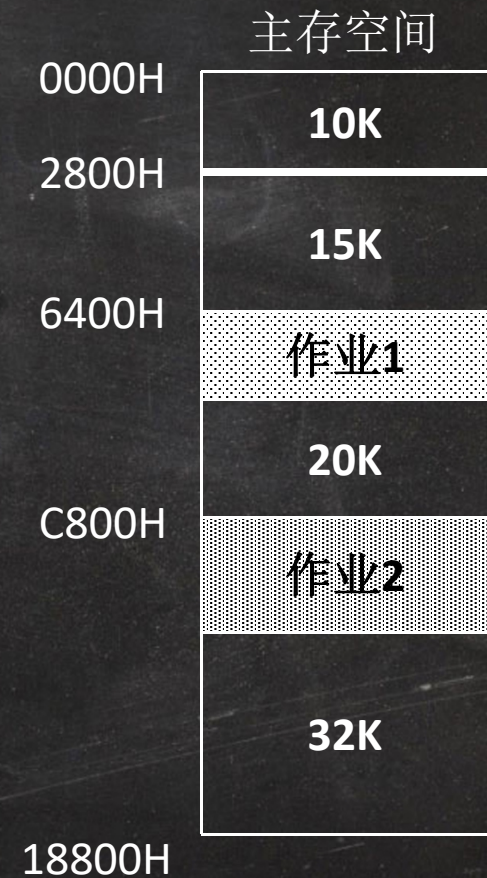
分区存储管理

- 把主存划分成若干个区域，每个区域分配给一个作业使用。这就是分区存储管理方式。分为固定分区、可变分区和可重定位分区。

(1) **固定分区**：系统生成时已经分好区。

(2) **可变分区**：是一种动态分区方式，存储空间的划分是在作业装入时进行的，故分区的个数是可变的，分区的大小刚好等于作业的大小。

(3) **可重定位分区**：分配好的区域可以移动。



分页存储管理

(1) 分页原理：

- 将进程的地址空间划分成若干个大小相等的区域，称为页。
- 将主存的空间也划分成与页相同大小的若干个物理块，称为块或页框。
- 在为进程分配主存时，将进程中若干页分别装入多个不相邻接的块中。

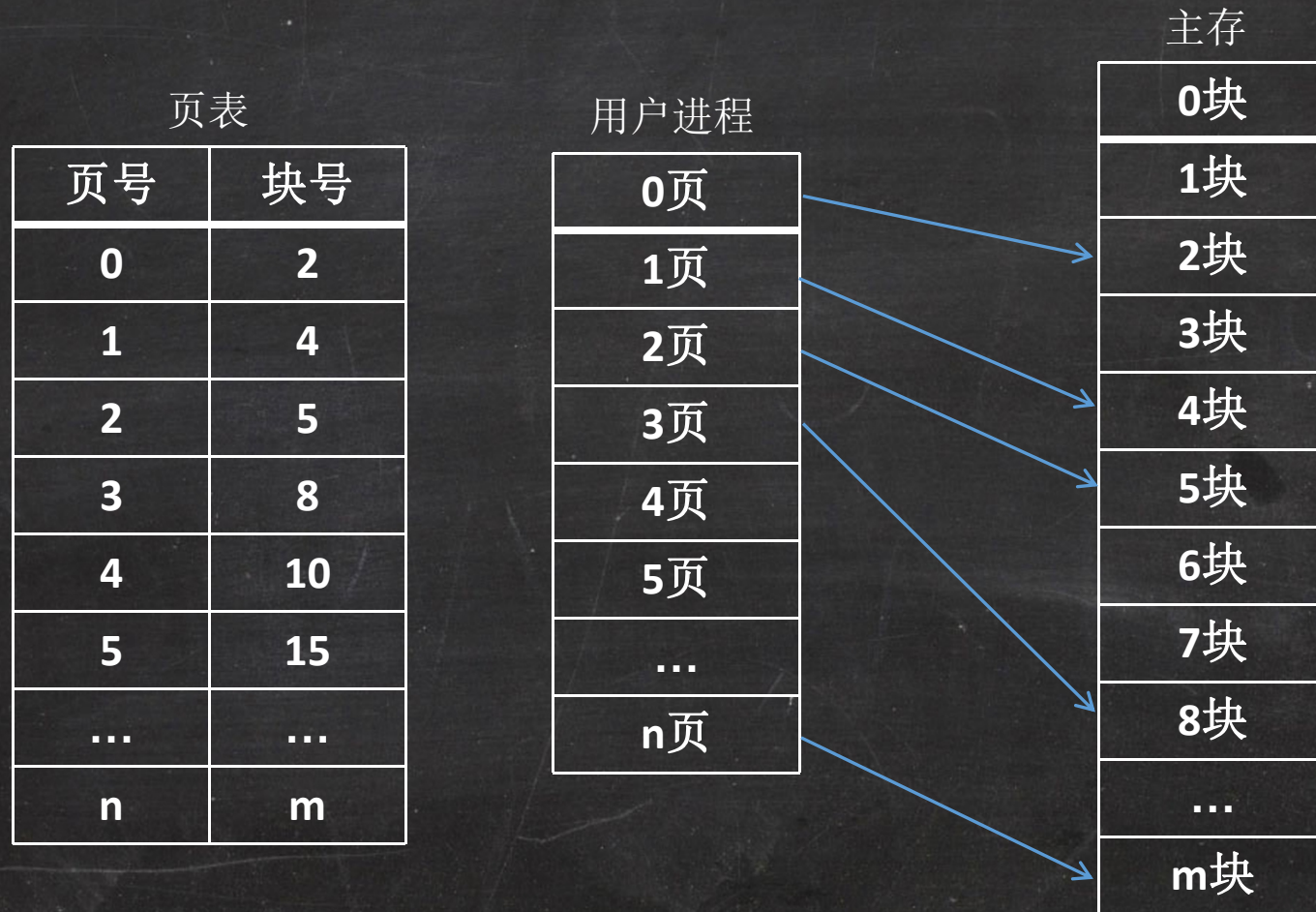
(2) 地址结构：



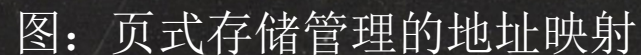
(3) 页表：

- 当进程的多个页面离散地分配到主存的多个物理块时，系统应能保证在主存中找到进程要访问的页面所对应的物理块，为此，系统为每个进程建立了一张页面映射表，简称页表。

分页存储管理

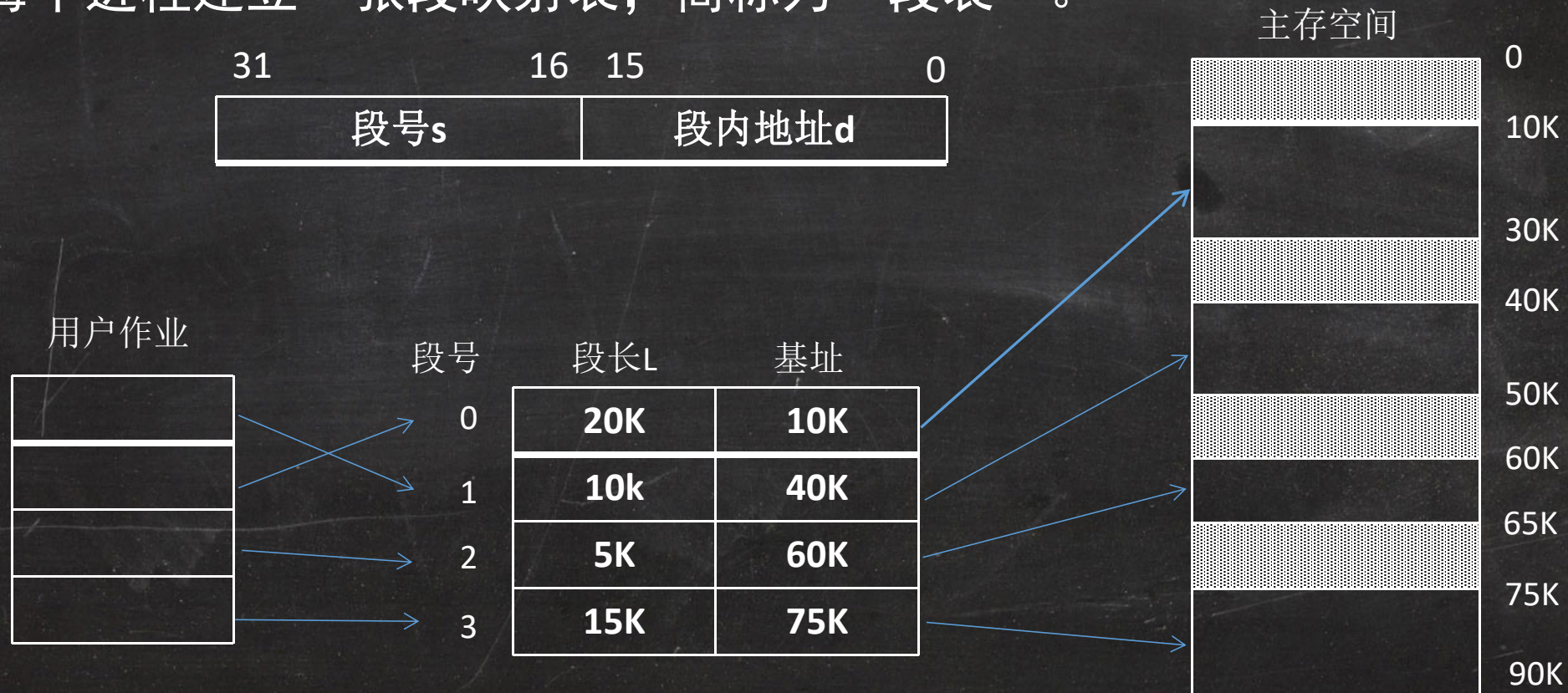


图：页表的地址映射

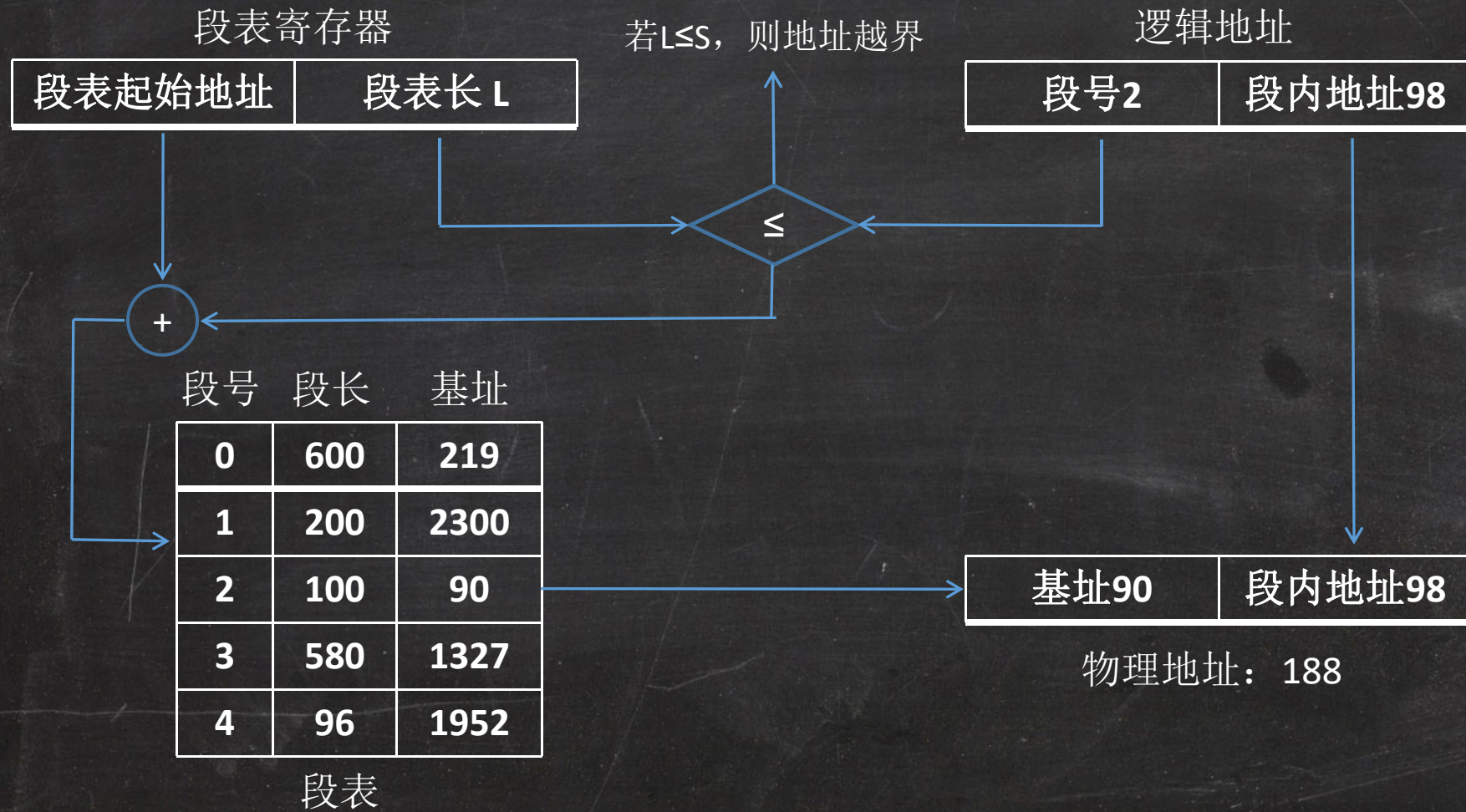


分段存储管理

- 在分段存储管理中，将用户程序或作业的地址空间按内容划分为段，比如主程序一段，子程序一段，数据专门放一段，每个段的长度是不等的，但是每个段占用一个连续的分区。进程中的各个段可以离散地分配到主存的不同分区中。在系统中为每个进程建立一张段映射表，简称为“段表”。



分段存储管理



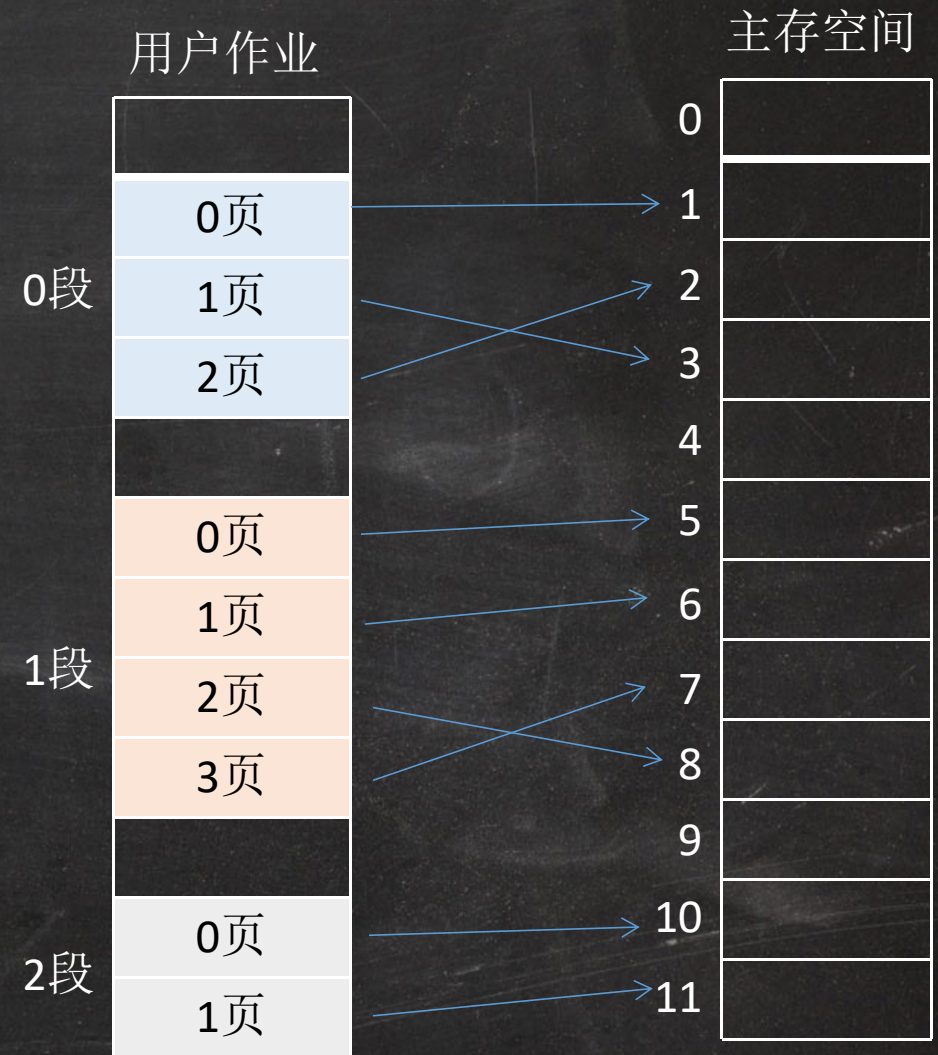
图：段式存储管理的地址变换

段页式存储管理

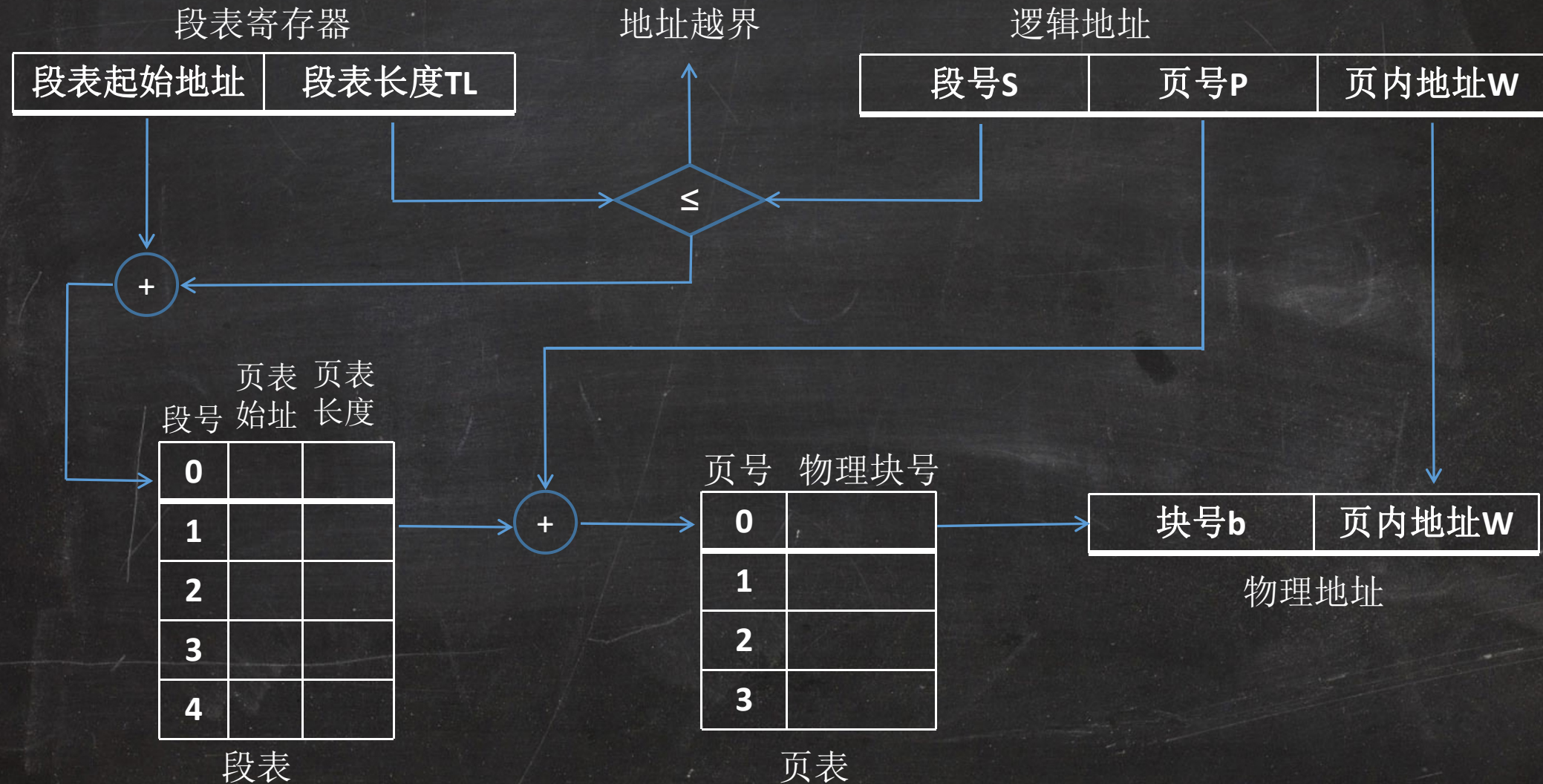
- 先将整个主存划分成大小相等的存储块（页框），将用户程序按程序的逻辑关系分为若干个段，然后再将段划分成页。

段页式系统中同时有段表和页表：

- **段表**：段号、页表始址、页表长度。
- **页表**：页号、物理块号。
- **逻辑地址**：段号、段内页号、页内地址
- **物理地址**：块号、页内地址



段页式存储管理



图：段页式存储管理的地址变换机构

虚拟存储管理

- **程序局部性原理：**程序在执行时将呈现出局部性规律，即在一段时间内，程序的执行仅限于某个部分，它所访问的存储空间也局限于某个区域内。
 - (1) 时间局限性：程序中的某条指令一旦执行，则不久的将来很有可能再次被访问；某个存储单元如果被访问，不久的将来它很可能再次被访问。
 - (2) 空间局限性：一旦程序访问了某个存储单元，则不久的将来，其附近的存储单元也最有可能被访问。
- 如果我们运行程序的时候，允许将作业的一部分装入主存即可运行程序，而其余部分可以暂时留在磁盘上，等需要的时候再装入主存。这样一来，一个小的主存空间就可以运行比它大的一个作业。从用户角度看，系统具有的主存容量比实际的主存容量要大得多，称为虚拟存储器。

请求分页系统的实现

- 在纯分页的基础上增加了请求调页功能，页面置换功能。
- 在请求分页系统中，每当所要访问的页面不在主存时便产生一个缺页中断。

例：在某计算机中，假设某程序的COPY指令跨两个页面，且源地址A和目标地址B所涉及的区域也跨两个页面，如下图所示：地址为A和B的操作数均不在内存，计算机执行COPY指令时，系统将产生（ ）次缺页中断；若系统产生3次缺页中断，那么该程序有（ ）个页面在内存。

A. 2 B. 3 C. 4 D. 5

A. 2 B. 3 C. 4 D. 5

B	
A	
指	copy A
令	to B

页面置换算法

- (1) **最佳置换算法**：是一种理想化的算法，即选择那些是永不使用的，或者是在最长时间不再被访问的页面置换出去。
- (2) **先进先出置换算法(FIFO)**：优先淘汰最先进入主存的页面，也就是在内存中停留时间最长的页面。
- (3) **最近最少未使用算法(LRU)**：优先淘汰最近这段时间用得最少的页面。系统为每一个页面设置一个访问字段，记录这个页面自上次被访问以来所经历的时间T，当要淘汰一个页面时，选择T最大的页面。
- (4) **最近未用置换算法(NUR)**：优先淘汰最近一段时间未引用过的页面。系统为每一个页面设置一个访问位，访问位为1代表访问过，为0代表没有被访问过，置换页面时选择访问位为0的置换出去。

1、17年第23题

某计算机系统页面大小为4K，进程的页面变换表如下所示。若进程的逻辑地址为2D16H。该地址经过变换后，其物理地址应为（ ）

A、2048H

B、4096H

C、4D16H

D、6D16H

页号	物理块号
0	1
1	3
2	4
3	6

2、15年第26题

某进程有4个页面，页号为0~3，页面变换表及状态位、访问位和修改位的含义如下图所示。若系统给该进程分配了3个存储块，当访问的页面1不在内存时，淘汰表中页号为（ ）的页面代价最小。

- A、0 B、1 C、2 D、3

页号	页帧号	状态位	访问位	修改位
0	6	1	1	1
1	—	0	0	0
2	3	1	1	1
3	2	1	1	0

状态位含义 { =0 不在内存
 =1 在内存

访问位含义 { =0 未访问过
 =1 访问过

修改位含义 { =0 未修改过
 =1 修改过

3、14年第26题

某计算机系统页面大小为4K，若进程的页面变换表如下所示，逻辑地址为十六进制1D16H。该地址经过变换后，其物理地址应为十六进制（ ）。

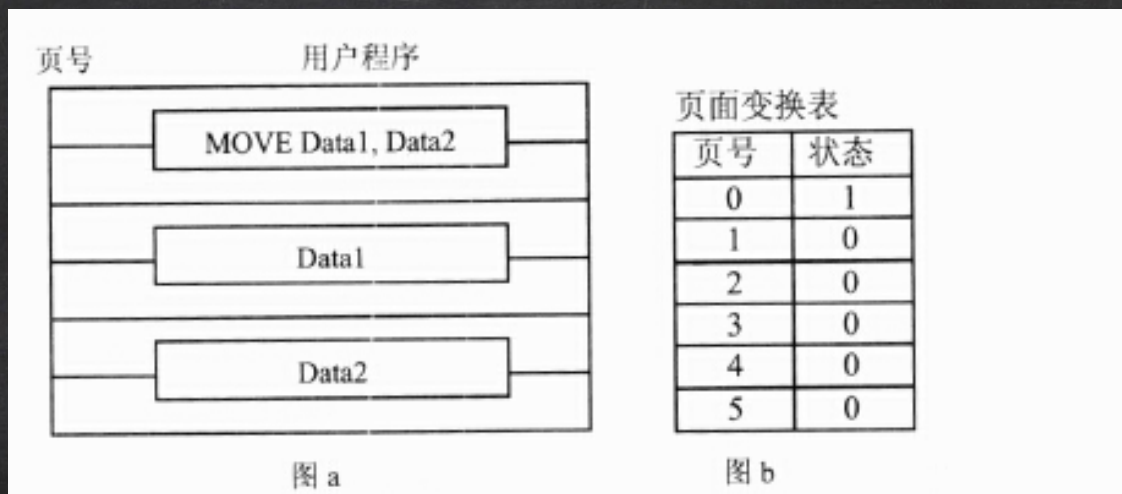
- A、1024H B、3D16H C、4D16H D、6D16H

页号	物理块号
0	1
1	3
2	4
3	6

4、13年第27题

假设内存管理采用可变式分区分配方案，系统中有五个进程P1~P5,且某一时刻内存使用情况如下图所示（图中空白处表示未使用分区）。此时，若P5进程运行完并释放其占有的空间，则释放后系统的空闲区数应（ ）

- A、保持不变 B、减1 C、加1 D、置零



分区号	进程
0	P1
1	P2
2	
3	P4
4	P3
5	
6	P5
7	

5、12年第26、27题

假设一台按字节编址的16位计算机系统，采用虚拟页式存储管理方案，页面的大小为2K，且系统中没有使用快表（或联想存储器）。某用户程序如图a所示，该程序的页面变换表如图b所示，表中状态位等于1和0分别表示页面在内存或不在内存。

图a中MOVE Data1,Data2是一个4字节的指令，Data1和Data2表示该指令的两个32位操作数。假设MOVE指令存放在2047地址开始的内存单元中，Data1存放在6143地址开始的内存单元中，Data2存放在10239地址开始的内存单元中，那么执行MOVE指令将产生（ ）次缺页中断，其中：取指令产生（ ）次缺页中断。

- A、3 B、4 C、5 D、6
A、0 B、1 C、2 D、3

6、11年第27题

某系统采用请求页式存储管理方案，假设某进程有6个页面，系统给该进程分配了4个存储块，其页面变换表如下表所示，表中的状态位等于1/0分别表示页面在/不在内存。当该进程访问的页面2不在内存时，应该淘汰表中页号为（ ）的页面。

A.0

B.3

C.4

D.5

页号	页帧号	状态位	访问位	修改位
0	5	1	1	1
1	—	0	0	0
2	—	0	0	0
3	2	1	1	0
4	8	1	1	1
5	12	1	0	0

【22年第20题】能够不访问页表，实现快速将虚拟地址映射到物理地址的硬件机制是（ ）。

- A.转换检测缓冲区 B.虚拟内存 C.多级页表 D.内存映射