第32讲 存储划分技术：简单分段技术

Simple Segmentation Technique

- 基于模块化程序设计时，程序员常根据需要将进程分割成许多大小不一定相同的 ***Segment***，系统则将 物理内存动态地划分成许多尺寸不一定相等的 ***Partition***.

- 当一个进程被装入物理内存时，系统将为该进程的每个段独立地分配一个分区；同一进程的多个段不必存放在连续的多个分区中 .

Simple Segmentation Technique

- All segments of all programs do not have to be of the same length.
- There is a maximum segment length.
- Logical address consists of two parts : a segment number and an offset.
- Since segments are not equal, segmentation is similar to dynamic partitioning.

Data Structure in Fragmentation

- **Segment Table(*段表*)**
  - 用于描述进程的分段情况，记载进程的各个段到物理内存中分区的映射情况。
  - 基本元素：段号、段的物理起始地址、段长度

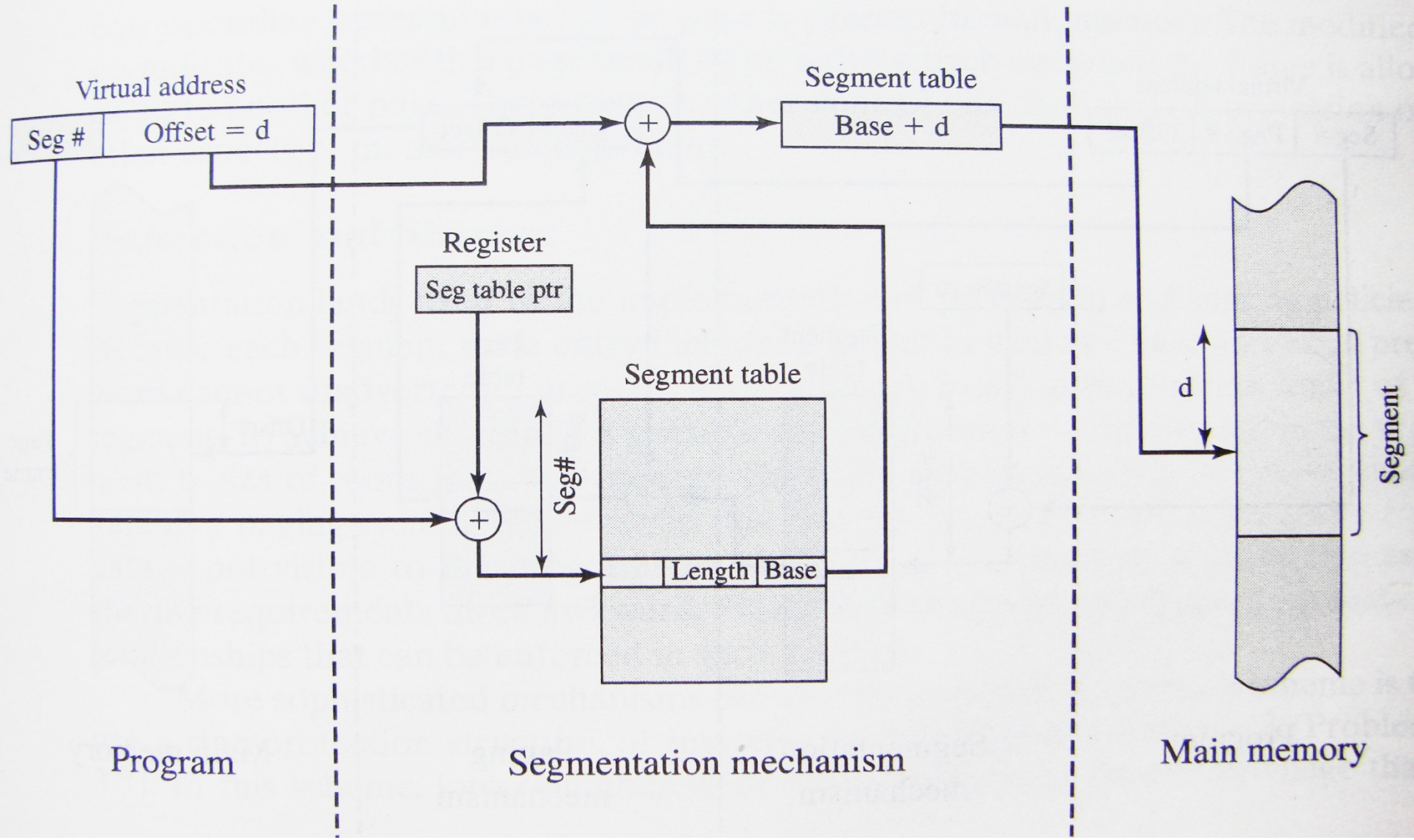
- **Partition Table(*分区表*)**
  - 用于记载物理内存的分区情况

分段系统中的地址变换和存储保护过程

- 在分段系统中， CPU 将按照以下步骤进行地址变换和存储保护：
  - 逻辑地址中的段号为索引检索段表从而得到指定段所对应的段表项；
  - 若逻辑地址中的段内偏移大于段表项中段的长度，则产生存储保护中断（该中断将由 OS 处理）；
  - 把逻辑地址中的段内偏移与段表项中的分区的起始物理地址相加从而得到物理地址

Segmentation Table Register

- 在分段系统中，段表被保存在物理内存中。
- **段表寄存器**是 CPU 中的一个硬件设施，用来存放当前正在运行的进程其段表在物理内存中的起始物理地址。
- 当系统调度某个进程运行时，它将从该进程的 PCB 中取出相应的值填入段表寄存器



Address Translation in Segmentation System

Protection and Sharing

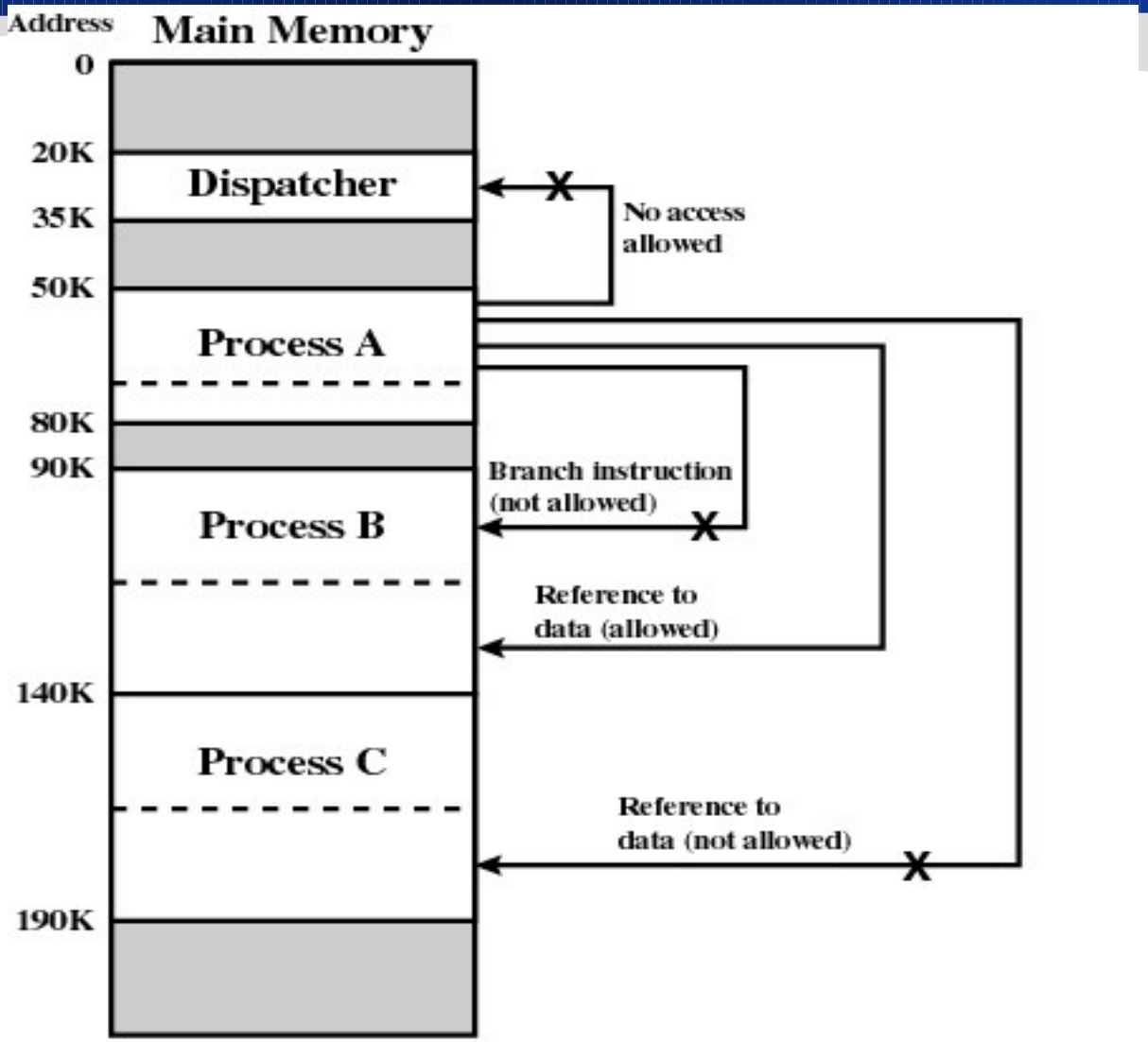


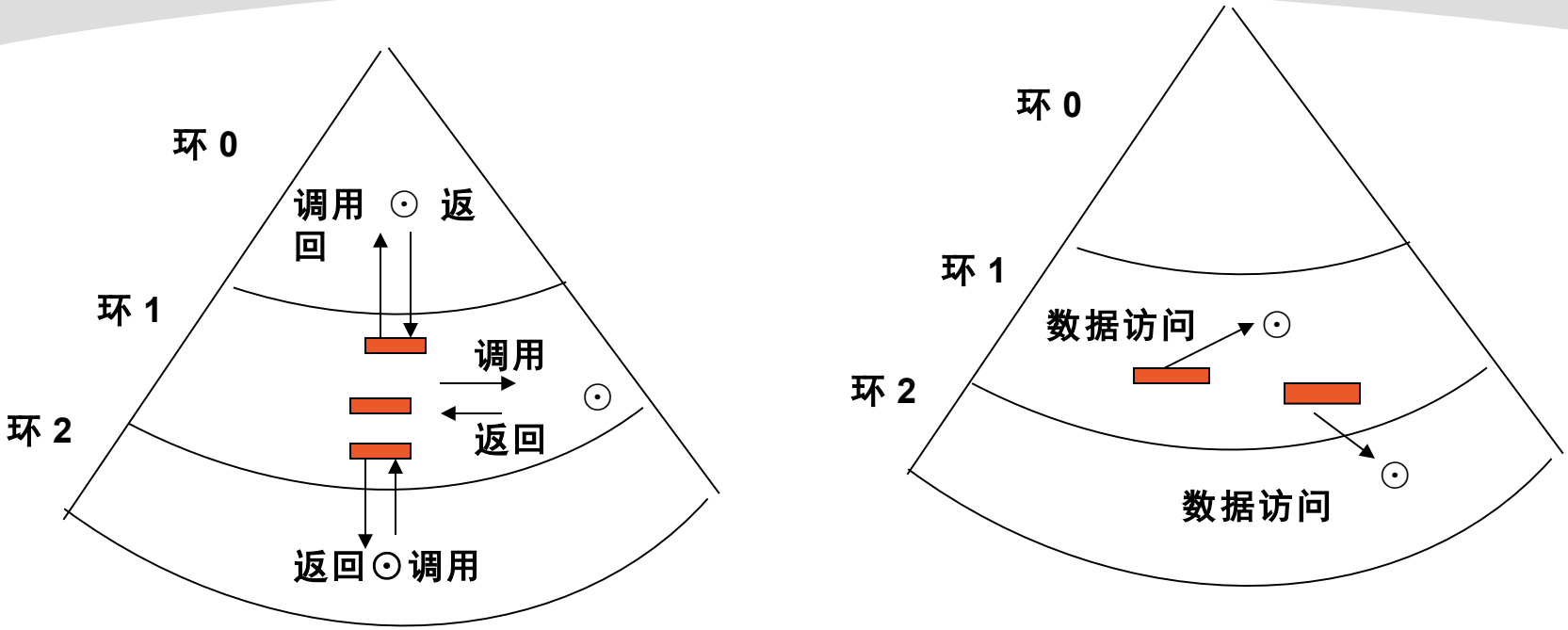
Figure 8.14 Protection Relationships Between Segments

Segment Protection

- ***越界检查***
  - 为了进行越界检查，系统在段表寄存器中保存当前进程的段表长度并在每个段表项中保存各个段的长度。
- ***存取控制***
  - 为了进行存取控制，系统在每个段表项中设置一个“存取控制”字段，用于规定进程对段的访问权限

Segment Protection

- 为了进行*环保护*，系统将在每个**段表项**中保存各个段所在环的特权级别。
- *环保护* 通常遵循以下保护规则：
  - 一个程序可以调用驻留在同级环或较高特权环中的服务；
  - 一个程序可以访问驻留在同级环或较低特权环中的数据



Ring protection structure

Paging *vs.* Segmentation

- 页面是信息的物理单位；在特定的系统中其大小是固定不变的，不随进程的不同而不同。
- 段是信息的逻辑单位，其长度不定；即使是属于同一进程的两个段其长度也可能不等。

Paging *vs.* Segmentation

- 分页活动源于系统管理物理内存的需要，在系统内部进行，由系统实施，用户看不见。

- 分段活动源于用户进行模块化程序设计的需要，在系统外部进行，由用户实施，用户是知道的。

Paging *vs.* Segmentation

- 在分页系统中，逻辑地址是一维的。
- 在分段系统中，逻辑地址是二维或多维的

Advantages of Segmentation/Paging

- 分页系统中，内零头得到了有效的抑制，外零头则完全被消除；因此，使用分页技术可以提高物理内存的利用率。

- 分段系统中，动态数据结构、程序和数据共享、程序和数据保护等问题得到了妥善解决；因此，分段技术有利于模块化程序设计。

- 段页技术汲取了分页技术和分段技术的上述优点

Combining Paging and Segmentation

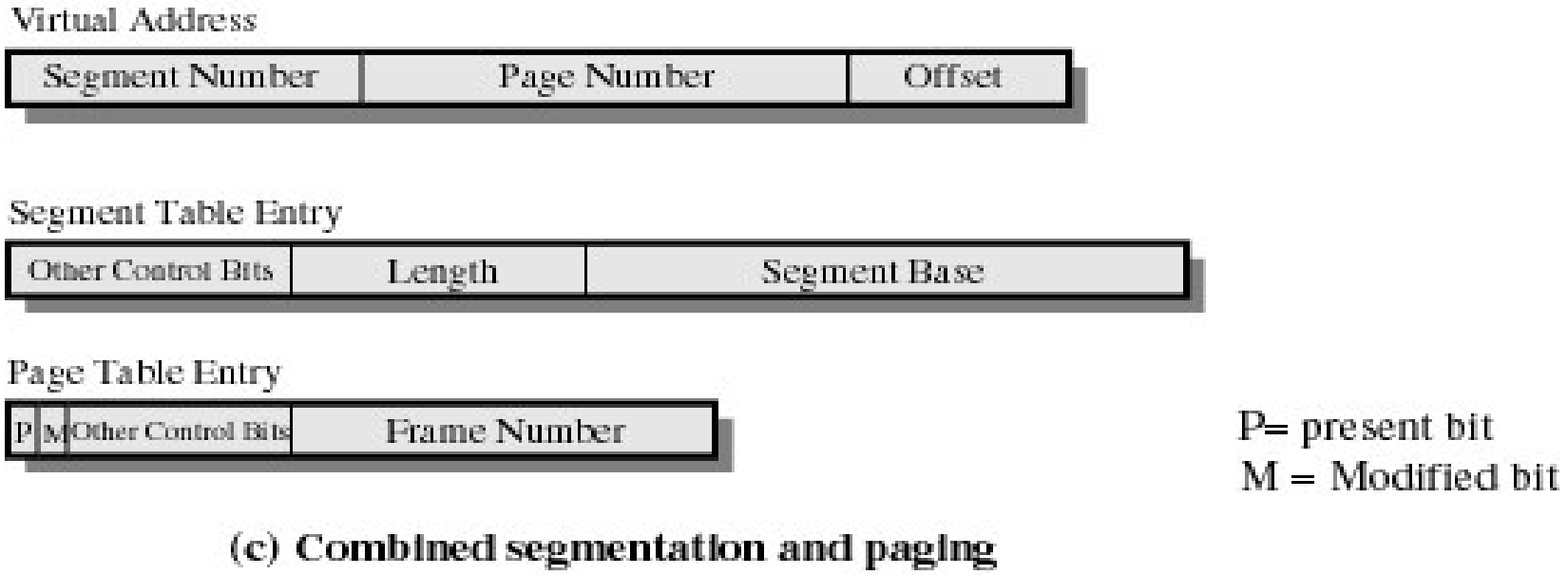
Combined Paging and Segmentation

- Paging is transparent to the programmer.
- Paging eliminates external fragmentation.
- Segmentation is visible to the programmer.
- Segmentation allows for growing data structures, modularity, and support for sharing and protection.
- Each segment is broken into fixed-size pages.

Combined Paging and Segmentation

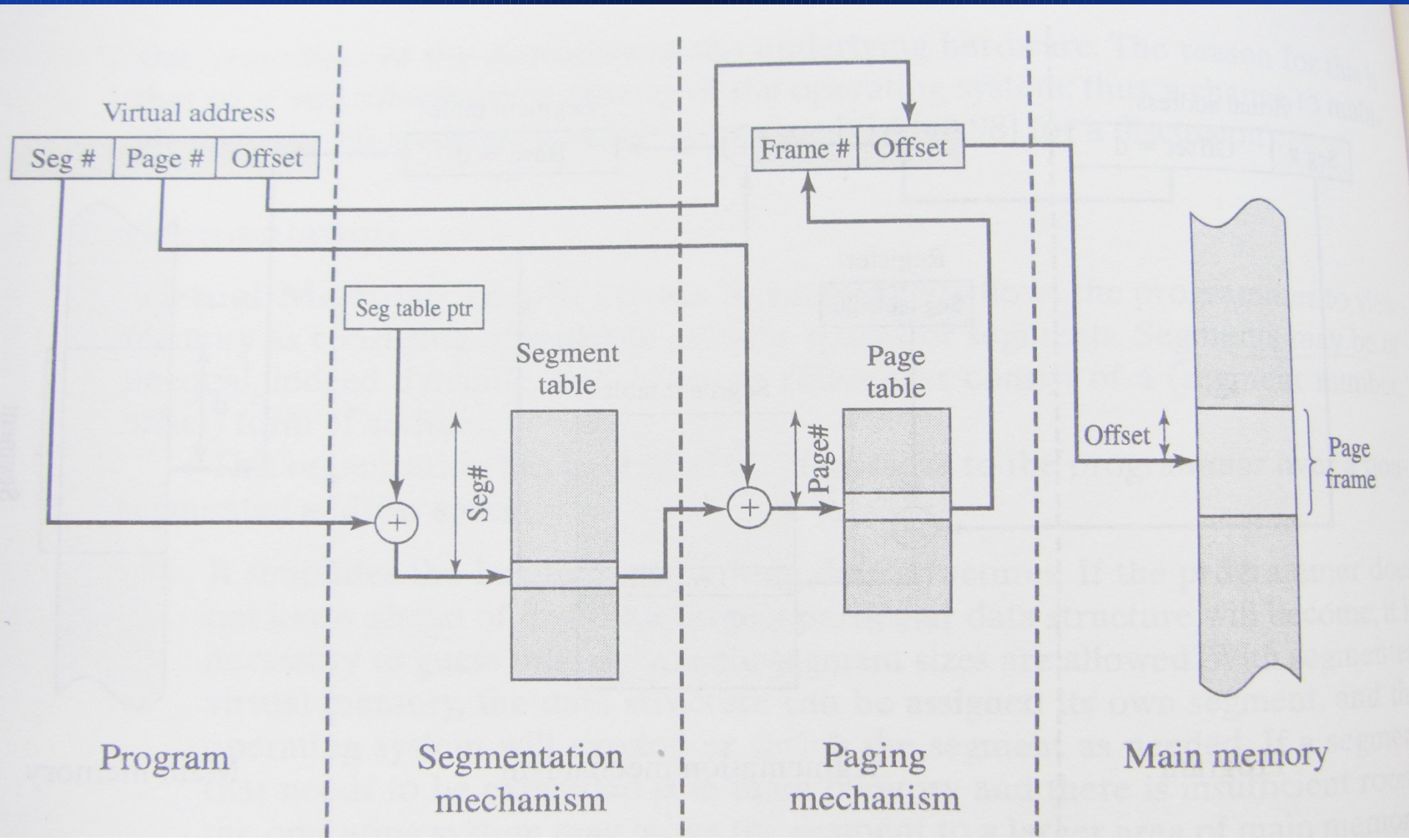
- 在段页存储管理系统中，每个进程均被编程人员分割成多个 *Segment*，　每个段又被系统分割成多个 *Page*。
- 相应地，物理内存被系统划分成多个 *Frame*。
- 当一个进程被装入物理内存时，系统为该进程的每个段的各页面独立地分配一个 *Frame*；一个进程的同一段的多个页面不必存放在连续的多个 *Frame* 中。

Combined Segmentation and Paging



(c) Combined segmentation and paging

Figure 8.2 Typical Memory Management Formats



Address Translation in Segmentation/paging System