程序装入的三种方式

- 绝对装入方式

  仅能运行单道程序，用户程序经过编译后，产生绝对地址（物理地址）的目标代码。

  绝对地址可由程序员直接给出，或在编译或汇编时给出。要求实现已知程序将装入内存的位置。

  优点：装入过程简单

  缺点：过于依赖硬件结构，不适于多道程序系统

- 可重定位装入方式（静态重定位）

  多道程序环境，逻辑地址映射到物理地址，地址变换在程序装入时一次完成，不可移动。

  物理地址=逻辑地址+内存中的起始地址

  优点：不需硬件支持，可以装入有限多道程序。

  缺点：一个程序通常需要占用连续的内存空间，程序装入内存后不能移动。不易实现共享。

- 动态运行时的装入方式（动态重定位）

  多道程序环境，逻辑地址映射到物理地址，地址变换在程序打算执行时完成，可移动。

  只有在程序真正执行到某一步时才对它进行地址转换。

  优点：OS可以将一个程序分散存放于不连续的内存空间，可以移动程序，有利于实现共享。

  缺点：该方式需要一定特殊硬件的支持，OS实现较复杂。

  是实现虚拟存储的基础。

链接的三种方式

- 静态链接

  程序运行之前，将各目标模块及它们所需的库函数连接成一个完整的可执行文件（装入模块），之后不再拆开。

- 装入时动态链接

  将各目标模块装入内存时，边装入边链接的链接方式。

- 运行时动态链接

  在程序执行中需要该目标模块时，才对它进行链接。便于修改和更新，便于实现对目标模块的共享。

存储管理的四个功能

地址转换（地址映射）

主存空间的存储保护

- 方法一：在CPU中设置一对上、下限寄存器，存放进程的上、下限地址。进程的指令要访问某个地址时，CPU检查是否越界。
- 方法二：采用重定位寄存器（基址寄存器）和界地址寄存器（限长寄存器）进行越界检查。重定位寄存器存放的是进程的**起始物理地址**，界地址寄存器中存放的是进程的**最大逻辑地址**。

主存空间的扩充

从逻辑上对内存空间进行扩充。

- 覆盖技术
    - 用来解决“程序大小超过物理内存总和”的问题。
    - 将程序分为多个段（多个模块）。

      常用的段常驻内存，不常用的段在需要时调入内存。

      内存中分为一个“固定区”和若干个“覆盖区”。需要常驻内存的段放在“固定区”中，调入后就不再调出（除非运行结束）。不常用的段放在“覆盖区”，需要用到时调入内存，用不到时调出内存。
    - 要求作业各模块之间有明确的调用结构，程序员要向系统指明覆盖结构，然后由操作系统完成自动覆盖。
    - 缺点：对用户不透明，增加了用户负担。
- 对换技术
    - 当内存空间紧张时，系统将内存中某些进程暂时移到外存，把外存中某些进程换进内存，占据前者所占用的区域。这种技术是进程在内存与外存之间的动态调度。
    - 多用于单道、时间片轮转的分时系统。
    - 磁盘空间分为文件区和对换区。文件区用于存放文件，追求存储空间的利用率，对文件区空间的管理采用离散分配方式；对换区只占磁盘空间的小部分，被换出的进程数据就存放在对换区。对换区的空间管理主要追求换入换出速度，采用连续分配方式。
    - 对换发生在进程或作业之间。
- 虚拟存储技术

  利用程序执行局部性原理，在磁盘上建立一个比实际主存空间大很多的地址空间用来装入程序。
    - 顺序局限性：某存储单元被访问过，后续单元因指令执行顺序，可能很快被访问。
    - 时间局限性：最近被访问过的迭代或循环可能被再次访问。
    - 空间局限性：相邻存储单元因数组堆栈可能被访问

  装入部分进程运行即可，提高处理器和主存空间的利用率。

主存空间的分配与回收

引起主存分配与回收的原因：

- 进程的开始和结束
- 进程运行过程中所占用的内存变化
- 进程映像在内存和外存之间传递
- 为了充分利用内存空间，系统可能对内存空间进行调整

连续分配管理方式

为用户程序分配一个连续的内存空间。

内部碎片：内存中已分配给作业但未被利用的区域称为“内部碎片”。

外部碎片：内存中的某些空闲分区由于太小而难以利用。

- 单一连续分配

  内存分为系统区、用户区。应用程序装入用户区，采用绝对地址，可使用用户区全部空间。

  内存中仅驻留一道程序，整个用户去为一个用户独占。

  - 易于管理，无外部碎片；可采用覆盖技术扩充内存；不一定需要采取内存保护。
  - 内存中只装入一道作业运行，有内部碎片；内存空间浪费大，各类资源的利用率也不高。

- 固定分区分配

  将整个用户空间划分为若干个固定大小的分区，在每个分区中只装入一道作业。划分主要在系统初始化时进行。

  - 分区大小相同：主要用于控制多个相同对象的场合。即各处理对象的大小基本相同。
  - 分区大小不同：一般可划分为多个小分区，适量中等分区，少量大分区。可适应多种类型的作业。
  - 分区使用表：将分区按大小顺序排列，并建立一张分区使用表。
  - 优点：实现简单，对于程序大小和出现频率已知的情形比较适合；无外部碎片
  - 缺点：预先规定分区大小，大作业可能无法装入；主存利用率不高，出现内部碎片；若一个进程在运行过程中要求动态扩充主存空间，较为困难；分区数目在系统初启动时确定，不适合分时交互或主存需求变化大的情形。

- 动态分区分配

  没有内部碎片，存在外部碎片。进程需要多少内存空间，就划分给它多少。

  - 空闲分区表
  - 空闲分区链

动态分区分配算法

算法	算法思想	分区排列顺序	优点	缺点
首次适应	从头到尾找适合的分区	空闲分区以地址递增次序排列	综合看性能最好。***算法开销小***，回收分区后一般不需要对空闲分区队列重新排序	
最佳适应	优先使用更小的分区，以保留更多大分区	空闲分区以容量递增次序排列	会有更多的大分区被保留下来，更能满足大进程需求	会产生很多太小的、难以利用的碎片；***算法开销大***，回收分区后可能需要对空闲分区队列重新排序
最坏适应	优先使用更大的分区，以防止产生太小的不可用的碎片	空闲分区以容量递减次序排列	可以减少难以利用的小碎片	大分区容易被用完，不利于大进程；***算法开销大***（原因同上）
邻近适应	由首次适应演变而来，每次从上次查找结束位置开始查找	空闲分区以地址递增次序排列（可排列成循环链表）	不用每次都从低地址的小分区开始检索。***算法开销小***（原因同首次适应算法）	会使高地址的大分区也被用完

离散分配方式

- 页式存储管理

  将程序的逻辑地址空间划分为固定大小的页，将物理内存划分为与页面大小相等的物理块（页框）。程序加载时，按页分配其所需的块，连续页面所分得的物理块页不必连续。需要CPU的硬件支持。

  最后一页装不满一块，称为页内碎片。

  - 不支持虚拟存储
  - 页面大小由硬件决定
  - 页表的功能：在页式存储管理系统中，允许将进程的每一页离散地存储在内存的任何一个物理页面上，为保证进程的正常运行，系统建立了页表，记录进程每一页被分配在内存的物理块号。页表的功能是实现从页号到物理块号的地址映射。
  - 地址空间很大时，页表也非常大，占有大量的内存空间。（例如：32位地址空间的页式系统，设页的大小为4KB，则进程的页表项最大可达1MB。若一个页表项占4B，则页表需要占用4MB的连续内存空间。）
  - 为解决这一问题可从两方面入手：一方面，可以将页表离散存储；另一方面可以将页表一部分调入内存，其余部分放在外存。

    具体实现方案：采用两级页表。页表分页，页面大小与内存物理块大小一致，并编号放入不同的物理块，离散分配的页表在建立一张页表，作为外层页表（页目录），此时进程的逻辑地址为： 外层页号+页号+页内位移。

    当运行进程，将外层页表调入内存，对所有页表而言，只调入少量的页表，使用时若找不到相应页表，则产生中断请求OS将需要的页表调入内存。两级页表适应了大地址空间的需要，由虚拟存储技术支持，但也增加了地址变换的开销和管理的复杂度。还可设计三级、四级页表。

  > 例：
  >
  > 为满足2^{64}地址空间的作业运行，采用多级分页存储管理方式。假设页面大小为4KB，在页表中的每个页表项要占8字节，则为了满足系统的分页管理至少应采用多少级页表?
  >
  > 页面大小4KB=2^{12}字节
  >
  > 每个页表项8字节=2^3字节
  >
  > 所以每个页面中可存放2^9个页表项
  >
  > ---
  >
  > 页面大小2^{12}字节，则页内偏移量占12位
  >
  > 剩余位数64-12=52位
  >
  > ---
  >
  > 每个物理块可存放2^9个页表项（和页面可存放的页表项一样）
  >
  > 所以需要52/9=6级页面

- 段式存储管理

5、分段与分页的主要区别（Main Difference）

- 1）页是信息的物理单位；分页目的：提高内存利用率；

  段是信息的逻辑单位；分段目的：满足用户需求。

- 2）页大小固定由系统确定；逻辑地址划分由硬件实现；

  段大小不定；逻辑地址划分由编译程序等软件完成。

- 3）分页的进程地址空间是一维的；

  分段的进程地址空间是二维的，在标识一个地址时，既要给出段名，也要给出段内地址。

- 4）通常段比页大，因而段表比页表短，可以缩短查找时间，提高访问速度。

- 分页管理
- 优点：
  - 解决了碎片问题
  - 便于管理
- 缺点：
  - 不易实现共享
  - 不便于动态连接

- 分段管理
- 优点：
  - 便于动态申请内存管理和使用统一化
  - 便于共享
  - 便于动态链接
- 缺点：产生碎片

- 段页式存储管理

  分段和分页相结合。**先**将用户程序分段，每段内**再**划分成若干页，每段有段名（段号），每段内部的页有一连续的页号。

  每个进程一张段表，每个段一张页表。

  段号S+段内页号P+页内地址W

虚拟存储管理

- 虚拟内存的最大容量：由计算机的地址结构（CPU寻址范围/地址长度）确定
- 虚拟内存的实际容量：MIN(内存和外存的容量之和，CPU寻址范围)

缺页中断属于内中断。

虚拟存储器的特征：离散性、多次性、对换性、虚拟性

页面置换算法

缺页率的影响因素：

- 页面大小。页面划分比较大，缺页率较低。
- 进程分配的物理块数。物理块越多，缺页率越低。

- 页面置换算法。算法的优劣决定进程执行过程中缺页中断的次数。
- 程序固有特性。程序本身的编制方法。程序局部化程度越高，缺页程度越低。

页面置换算法：

- 最佳置换OPT：最理想的算法，缺页率低。
- 先进先出FIFO
- 最近最久未使用LRU
- Clock置换（最近未用算法NRU）：为每页设置一访问位，链成一循环队列。寻找访问位为0的页面作为被置换位。
    - 改进的Clock：置换既未使用又未修改的页面。设置访问位A和修改位M。先找A=0,M=0,再找A=0,M=1
- 最不常用LFU
- 页面缓冲PBA

	算法规则	优缺点
OPT	优先淘汰最长时间内不会被访问的页面	缺页率最小，性能最好；但无法实现
FIFO	优先淘汰最先进入内存的页面	实现简单；但性能很差，可能出现Belady异常
LRU	优先淘汰最近最久没访问的页面	性能很好；但需要硬件支持，算法开销大
CLOCK（NRU）	循环扫描各页面 第一轮淘汰访问位=0的，并将扫描过的页面访问位改为1。若第一轮没选中，则进行第二轮扫描。	实现简单，算法开销小；但未考虑页面是否被修改过。
改进型CLOCK（改进型NRU）	若用（访问位,修改位）的形式表述，则 第一轮：淘汰（0,0） 第二轮：淘汰（0,1），并将扫描过的页面访问位都置为0 第三轮：淘汰（0,0） 第四轮：淘汰（0,1）	算法开销较小，性能也不错

习题

1. 在页式管理中，页表的起始地址存放在：
   A. 内存
   B. 存储页面表中
   C. 联想存储器中
   D. 寄存器中
2. 某系统采用页式存储管理策略，拥有逻辑空间64页，每页2KB，则用来描述页内地址的位数和页号的位数分别是：
   A. 11,5
   B. 10,5
   C. 10,6
   D. 11,6

3. 某分页存储管理的系统中，内存容量为1MB，被分成256块，块号为0、1、...、255，则作业每一页的长度为：
   A. 4KB
   B. 3KB
   C. 2KB
   D. 1KB

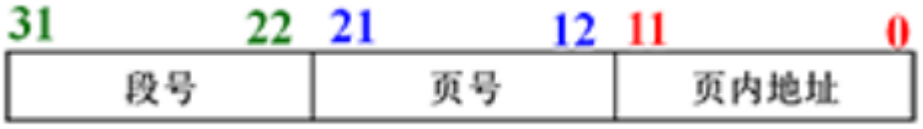
4. 下面关于请求分段存储管理的叙述中说法正确是：
   A. 分段尺寸受内存空间的限制，且作业总的尺寸也受内存空间的限制。
   B. 分段尺寸不受内存空间的限制，但作业总的尺寸受内存空间的限制。
   C. 分段尺寸不受内存空间的限制，且作业总的尺寸不受内存空间的限制。
   D. 分段尺寸受内存空间的限制，但作业总的尺寸不受内存空间的限制。

5. 段式存储管理中，分段是由用户决定的，因此:
   A. 段内地址和段间的地址都是连续的
   B. 段内地址是连续的，而段间的地址是不连续的
   C. 段内地址是不连续的，而段间的地址是连续的
   D. 段内地址和段间的地址都是不连续的

6. 以下解决主存碎片问题较好的存储器管理方式是：
   A. 可变式分区
   B. 分页管理
   C. 分段管理
   D. 单一连续区管理

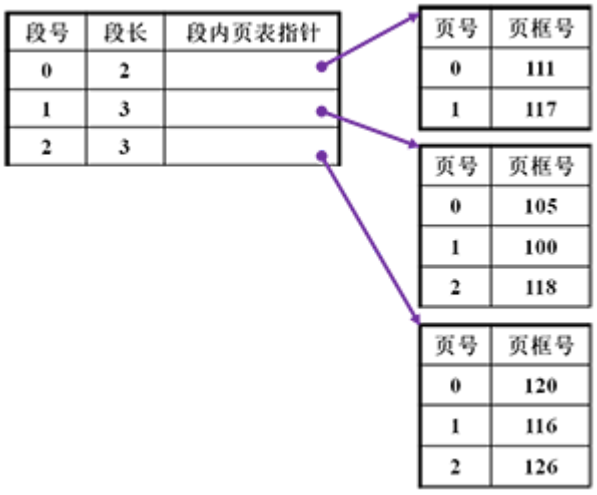
7. 段页式存储管理中，地址映射表是（　）
   A. 每个进程一张段表，两张页表
   B. 每个进程的每个段一张段表，一张页表
   C. 每个进程一张段表，每个段一张页表
   D. 每个进程一张页表，每个段一张段表

8. 假设段页式存储管理系统中的地址结构如下图所示，则系统：

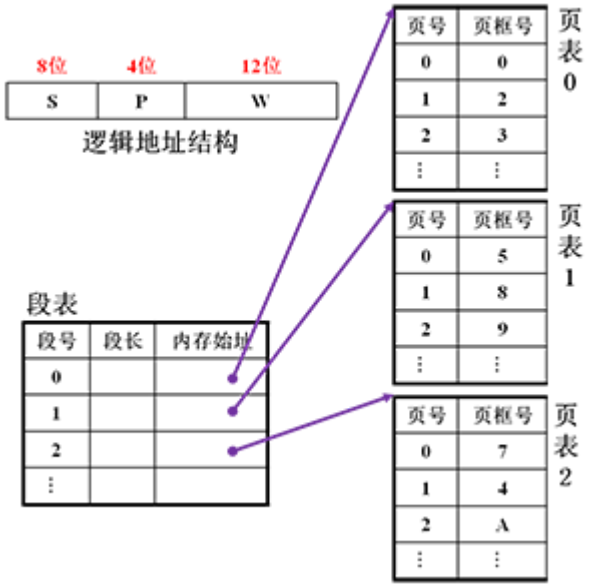


   A. 最多可有2048个段，每个段的大小均为2048个页，页的大小为2K
   B. 最多可有2048个段，每个段最大允许有2048个页，页的大小为2K
   C. 最多可有1024个段，每个段的大小均为1024个页，页的大小为4K
   D. 最多可有1024个段，每个段最大允许有1024个页，页的大小为4K

9. 段页式存储管理中，某个进程的段表和页表如下图所示，页的大小为4096B，现有逻辑地址(2，8976)，求其对应的物理地址。



10. 某系统采用段页式存储管理，有关数据结构如下图所示，计算逻辑地址139366对应的物理地址。



11. 说明缺页中断与一般中断的主要区别。

12. 局部置换和全局置换有什么区别？多道程序系统中，建议选择哪种置换？

13. 在请求分页系统的页表中增加了若干项，其中状态位供 空1 参考；修改位供 空2 时参考；访问位供 空3 参考；外存始址供 空4 参考。（空中填对应的字母即可）

    A.分配页面 B.置换算法 C.程序访问 D.换出页面 E.调入页面

14. 在请求页式存储管理系统中，页面大小为100B，有一个50*50的数组按行连续存放，每个整数占2B。将数组初始化的程序如下：

    程序A：

```
int i,j;
int a[50][50];
for (i=0; i<50; i++)
  for (j=0; j<50; j++)
        a[i][j]=0;
```

程序B：

```
int i,j;
int a[50][50];
for (j=0; j<50; j++)
    for (i=0; i<50; i++)
        a[i][j]=0;
```

若程序执行过程中，内存中只有一个页面用来存放数组的信息，试问：

(1)程序A执行时产生的中断次数是

(2)程序B执行时产生的中断次数是

15. 某虚拟存储器的用户空间共64个页面，每页的大小为1K，一个进程的大小占5个页面，系统为它分配了3个物理块。当前进程的页表如图所示：

页号	块号	存在位 P	访问位 R	修改位 M
0	1C	1	1	0
1	3F	1	1	1
2	–	0	0	0
3	5D	1	0	0
4	–	0	0	0

(1)计算逻辑地址的有效位是多少位?

(2)该进程的哪些页面不在内存?

(3)请分别计算进程中虚地址为03C7(H)， 12A8(H)， 1543(H)单元的物理地址（用十六进制表示），并说明理由。

*提示，当有缺页中断时，按改进Clock算法查看访问位和修改位，决定换出页。

答案

DDADB BCD

9. 逻辑地址(2，8976)可知段号为2，页地址为8976；

   页的大小为4096B，所以根据页地址8976可计算页号和页内地址为：
   页号=8976 / 4096 =2
   页内地址= 8976 % 4096 =784
   由段号2查到其对应的页表，第2页对应的页框号为126，因此可计算物理地址为：
   126×4096+784=516880

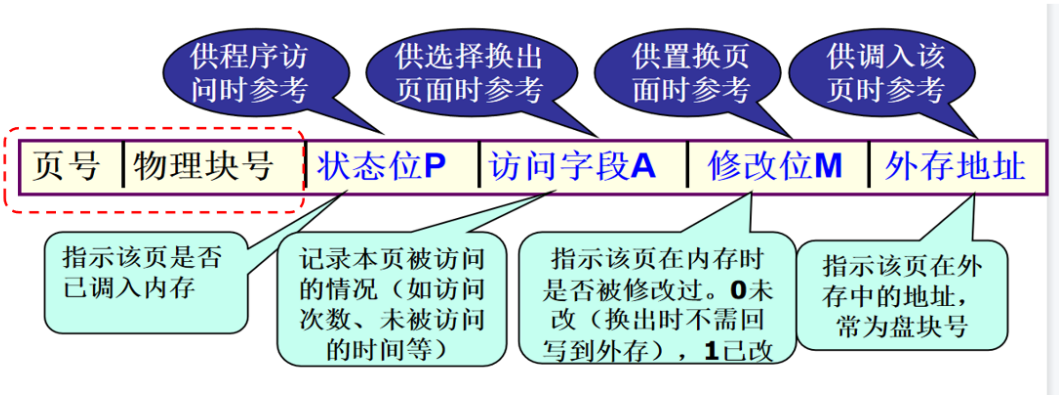
10. 逻辑地址139366的二进制为：10 0010 000001100110，将则持分为下列形式：
    页内地址w为12位，可知页大小为2^12=4096，给定的逻辑地址中页内地址为：
    000001100110，十进制为：102
    页号P为4位，即：0010，十进制为：2
    剩余的为段号S（8位），即：10，十进制为：2
    查段表的第2号指向的页表，可得第2页对应的页框号为A，即十进制10，因此物理地址为：
    10×4096+102=41062

11. 缺页中断与一般中断一样要经历保护CPU现场，分析中断原因，转中断处理程序进行处理及恢复中断现场等步骤，但缺页中断是一种特殊的中断，区别主要两个：
    (1)在指令执行期间产生和处理中断。 通常CPU是一条指令执行后检查是否有中断发生，若有去处理；否则执行下一条指令。但缺页中断在指令执行期间发现要访问的指令或数据不在内存时，产生和处理中断。

(2)一条指令执行期间可能产生多次中断。如要求读取多个字节数据的指令，指令中的数据可能跨多个页面，该指令执行可能发生多次中断：访问指令中断、访问数据中断。

12. 局部置换：当进程在执行过程中发生缺页时，只在分配给该进程的物理块中选择一页换出。
    全局置换：在所有用户能使用的整个存储空间中选择一个页面换出。
    建议：局部置换策略。 因为即使某个程序出现抖动现象，不会引起其他进程产生抖动，将抖动控制在最小范围内。

13.

14. 

重点注意访问字段，供换出页面参考。

14. 50 2500

15. (1) 64=2^6,即逻辑地址页号数共6位；1K=2^10,即页内地址数10位；共16位
    (2)该进程的第2页和第4页不在内存。
    (3)

    03C7H = (0000 0011 1100 0111)2，故该虚地址对应的页号是0，该页在内存，页内地址为(11 1100 0111)2。查表可知，0号页对应的块号为1C = (0001 1100)2。因此，对应的物理地址为：(0001 1100 11 1100 0111)2 = (0111 0011 1100 0111)2 = 73C7H。

    12A8 = (0001 0010 1010 1000)2，故该虚地址对应的页号是4，缺页，发生缺页中断，根据访问位，淘汰第3页，第4页装入块号为5D的物理块中。页内地址为(10 1010 1000)2。块号5D = (0101 1101)2。因此，对应的物理地址为：(0101 1101 10 1010 1000)2 = (0001 0111 0110 1010 1000)2 = 176A8H

    1543 = (0001 0101 0100 0011)2，故该虚地址对应的页号是5，发生越界中断。