计算机考研系列书课包

玩转操作系统

| 主讲人 | 刘财政

第四讲 内存管理

本讲内容

考点一: 内存管理概述

考点二: 内存连续分配管理方式

考点三: 基本分页内存管理方式

考点五: 虚拟存储器

考点五: 虚拟存储器

考点五: 带快表的两级系统的地址翻译过程

*

* ***

考点四:

基本分段内存管理方式



分段系统的基本原理



分段的基本结构



段表



地址变换机构

分段系统的基本原理

引入分段存储管理方式, 主要是为了满足用户和程序员的下述一系列需要:

- □ 方便编程
- □ 信息共享
- □ 信息保护
- □ 动态增长
- □ 动态链接

□方便编程

- 用户把自己的作业按照逻辑关系划分为若干个段,每个段都是从0开始编址, 并有自己的名字和长度。
- 要访问的逻辑地址是由段名(段号)和段内偏移量(段内地址)决定的。
- 例如,下述的两条指令便是使用段名和段内地址:
- LOAD 1, [A] | \langle D \rangle;
- STORE 1, [B] | (C);
- 其中, 前一条指令的含义是将分段A中D单元内的值读入寄存器1;
- 后一条指令的含义是将寄存器1的内容存入B分段的C单元中。

□ 信息共享

- 在实现对程序和数据的共享时,是以信息的逻辑单位为基础的。比如,共享变量、某个例程或函数。
- 分页系统中的"页"只是存放信息的物理单位(块),并无完整的意义,不便于实现共享;然而段却是信息的逻辑单位。
- 为了实现段的共享,希望存储管理能与用户程序分段的组织方式相适应。

□ 信息保护

 信息保护是对信息的逻辑单位进行保护,因此,分段管理方式能更有效和方便 地实现信息保护功能。

□ 动态增长

- 在实际应用中,有些段,特别是数据段,在使用过程中会不断地增长,而事先又无法确切地知道数据段会增长到多大。
- 分段存储管理方式却能较好地解决这一问题。

分段的基本结构

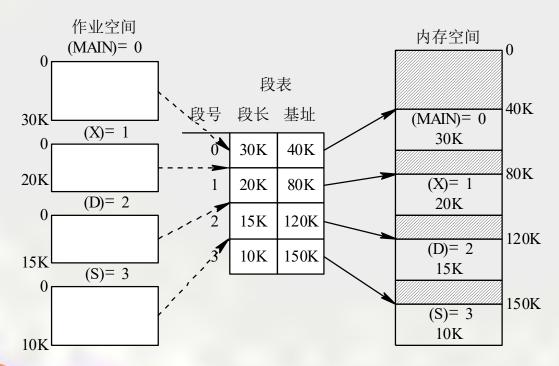
分段地址中的地址具有如下结构:

	段号			段内地址	
31		16	15		0

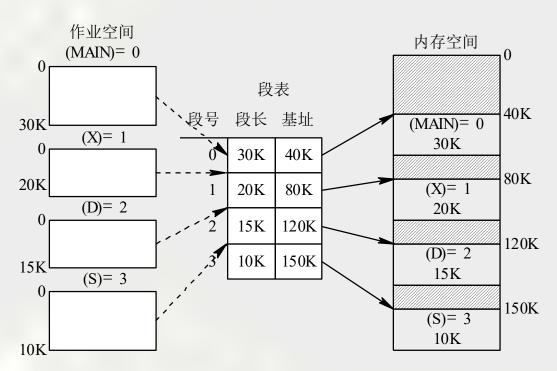
- 在分段存储管理方式中,作业的地址空间被划分为若干个段,每个段是一组完整的逻辑信息
- □ 每个段都有自己的名字(段号),都是从零开始编址的一段连续的地址空间, 各段长度是不等的。
- □ 内存空间被动态的划分为若干个长度不相同的区域, 称为物理段, 每个物理段由起始地址和长度确定

段表

- □ 在分段式存储管理系统中,则是为每个分段分配一个连续的分区,而进程中的 各个段可以离散地移入内存中不同的分区中。
- □ 为能从物理内存中找出每个逻辑段所对应的位置,在系统中为每个进程建立一 张段映射表,简称"段表"。

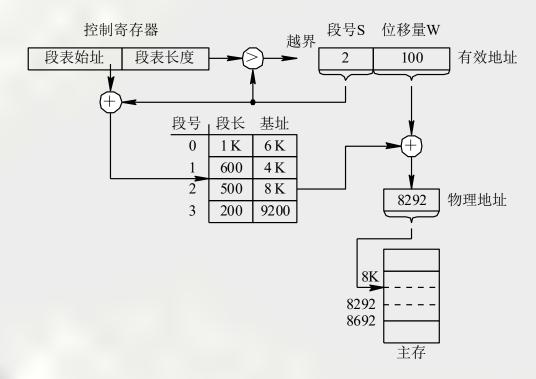


□ 每个段在表中占有一个表项,其中记录了该段在内存中的起始地址(又称为"基址")和段的长度,



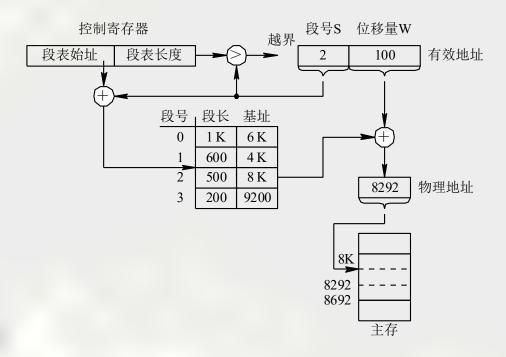
地址变换机构

□ 为了实现进程从逻辑地址到物理地址的变换功能,在系统中设置了<mark>段表寄存器</mark>,用于存放<mark>段表始址和段表长度 TL</mark>。在进行地址变换时,如图所示,



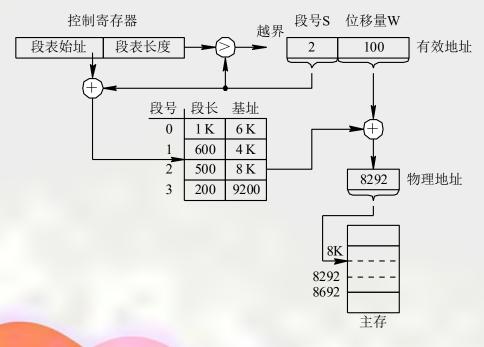
在进行地址变换时,如图所示,

- □ 系统将逻辑地址中的<mark>段号与段表长度 TL</mark> 进行比较。
- □ 若 S>=TL,表示段号太大,是访问越界,于是产生越界中断信号。



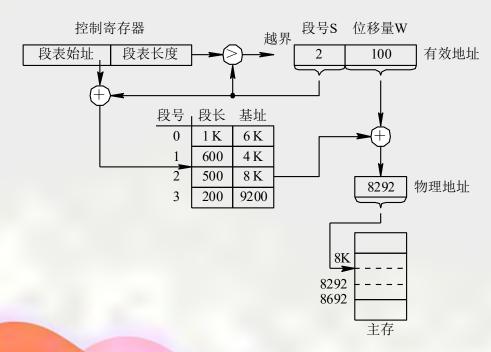
在进行地址变换时,如图所示,

- □ 若未越界,则根据段表的始址和该段的段号,计算出该段对应<mark>段表项</mark>的位置,从中读出该段在内存的起始地址。
- □ 再检查段内地址 d 是否超过该段的段长SL。
- □ 若超过,即 d>SL,同样发出越界中断信号。



在进行地址变换时,如图所示,

- □ 若未越界,则根据段表的始址和该段的段号,计算出该段对应<mark>段表项</mark>的位置,从中读出该段在内存的起始地址。再检查段内地址 d 是否超过该段的段长SL。
- □ 若超过,即 d>SL,同样发出越界中断信号。
- □ 若未越界,则将该段的基址 与段内地址d相加,即可得到要访问的内存物理地址



页是信息的物理单位

页的大小固定且由系统固定

分页的用户程序地址空间是一维的

分段和分页的区别

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。

分页是系统管理的需要,分段是用户应用的需要。一条指令或一个操作数可能会跨越两个页的分界处,而不会跨越两个段的分界处。

【政哥点拨】

址所对应的物理地址。

段号	内存始址	段长
0	210	500
1	2350	20
2	100	90
3	1350	590
4	1938	95

表1段表

段号	段内位移
0	430
1	10
2	500
3	400
4	112
5	32

表1 逻辑地址

所对应的物理地址。

段号	内存始址	段长
0	210	500
1	2350	20
2	100	90
3	1350	590
4	1938	95

表1 段表

段号	段内位移
0	430
1	10
2	500
3	400
4	112
5	32

表1 逻辑地址

【解析】 (1)由段表知,第0段内存始址为210,段长为500,因此逻辑地址(0,430)是合法地址,对应的物理地址为210+430=640。

所对应的物理地址。

段号	内存始址	段长
0	210	500
1	2350	20
2	100	90
3	1350	590
4	1938	95

表1 段表

段号	段内位移
0	430
1	10
2	500
3	400
4	112
5	32

表1 逻辑地址

(2)由段表知,第1段内存始址为2350,段长为20,因此逻辑地址(1,10)是合法地址,对应的物理地址为2350+10=2360。

所对应的物理地址。

段号	内存始址	段长
0	210	500
1	2350	20
2	100	90
3	1350	590
4	1938	95

表1 段表

段号	段内位移
0	430
1	10
2	500
3	400
4	112
5	32

表1 逻辑地址

(3)由段表知,第2段内存始址为100,段长为90,逻辑地址(2,500)的段内位移500超过了段长,因此为非法地址。

所对应的物理地址。

段号	内存始址	段长
0	210	500
1	2350	20
2	100	90
3	1350	590
4	1938	95

表1 段表

段号	段内位移
0	430
1	10
2	500
3	400
4	112
5	32

表1 逻辑地址

(4)由段表知,第3段内存始址为1350,段长为590,因此逻辑地址(3,400)是合法地址,对应的物理地址为1350+400=1750。

所对应的物理地址。

段号	内存始址	段长
0	210	500
1	2350	20
2	100	90
3	1350	590
4	1938	95

表1 段表

段号	段内位移	
0	430	
1	10	
2	500	
3	400	
4	112	
5	32	

表1 逻辑地址

(5)由段表知,第4段内存始址为1938,段长为95,逻辑地址(4,112)的段内位 移112超过了段长,因此为非法地址。

所对应的物理地址。

段号	内存始址	段长
0	210	500
1	2350	20
2	100	90
3	1350	590
4	1938	95

表1段表

段号	段内位移	
0	430	
1	10	
2	500	
3	400	
4	112	
5	32	

表1 逻辑地址

(6)由段表知,不存在第5段,因此逻辑地址(5,32)为非法地址。

2. 采用分段存储管理的系统中, 若段地址用24位表示, 其中8位表示段号, 则允 许每段的最大长度是()。

A. 2²⁴B B. 2¹⁶B C. 2⁸B D. 2³²B

分段地址中的地址具有如下结构:

	段号			段内地址	
31		16	15		0

B【解析】段地址为24位,其中8位表示段号,则段内偏移量(段内位移)占用剩 余的16位,因此最大段长为216 B。

3. 某进程的段表内容如表所示。

表 某进程的段表

段号	段长	内存起始地址	权限	状态
0	100	6000	只读	在内存
1	200	_	读写	不在内存
2	300	4000	读写	在内存

当访问段号为2、段内地址为400的逻辑地址时,进行地址转换的结果是()。

- A. 段缺失异常 B. 得到内存地址4400
- C. 越权异常 D. 越界异常
- D【解析】访问段号2,找到段表第三行段表项,段内地址为400,与段长比较发现大于段长,故地址越界异常,选D。

【牛刀小试】

- 1. 在以下存储管理方式中,会产生内部碎片的是()。
 - Ⅰ. 分段虚拟存储管理 Ⅱ. 分页虚拟存储管理
 - Ⅲ. 段页式分页管理 IV. 固定式分区管理
 - A. 仅 I 、 **工**和**田** B. 仅**田**和**I**V
 - C. 仅I D. 仅I、II和IV
- D【解析】只要是固定的分配就会产生内部碎片,其余的都会产生外部碎片。如果固定和不固定同时存在(例如段页式),还是看成固定的。

分段虚拟存储管理:每一段的长度都不一样(对应不固定),所以会产生外部碎片;

分页虚拟存储管理:每一页的长度都一样(对应固定),所以会产生内部碎片;

段页式分区管理: 既有固定, 也有不固定, 以固定为主, 所以会产生内部碎片;

固定式分区管理: 很明显固定, 会产生内部碎片。

综上分析,Ⅱ、Ⅲ、Ⅳ会产生内部碎片。因此选择D。

- 2. 【中国计量大学 2020】在基本分段存储管理中,逻辑地址转换为物理地址时,若段号超过段表长度,则会引起()。
 - A. 输入输出I/O中断 B. 缺段中断
 - C. 越界中断 D. 缺页中断
- C【解析】判断段号是否越界的依据:逻辑地址的段号S与段表寄存器的段表长度 TL进行比较(段号S≥段表长度TL就是越界)。段表长度至少是1,而段号从0开始。若段号超过段表长度,则会引起越界中断。因此选择C。

- 3. 【吉林大学 2014】段页式存储管理的逻辑地址形式为(s,p,d), 其中可能越界 的成分有()。
 - A. s, p和d B. s和p C. s和d D. p和d

- B【解析】段页式存储管理的逻辑地址形式为(s, p, d), 其中可能越界的成分有 段号s和逻辑页号p。因此选择B。

- 4. 【南京工业大学 2013】采用段页式存储管理的系统中,若地址用32位表示,其中10位表示段号,页的大小为4 KB,则允许每段的最大页号是()。
 - A. 1024 B. 1023 C. 4096 D. 4095
- B【解析】页的大小为4KB,则页内地址是12位,由于10位表示段号,则页号占用的位数是10位,则允许每段的最大页号为1023。因此选择B。

- 5. 采用分段式存储管理时,一个程序如何分段是在()时决定的。

- A. 分配主存 B. 用户编程 C. 装作业 D. 程序执行
- B【解析】分段是在用户编程时,将程序按照逻辑划分为几个逻辑段。因此选择B。
- 引入分段存储管理方式, 主要是为了满足用户和程序员的下述一系列需要:
- □ 方便编程
- □ 信息共享
- □ 信息保护
- □ 动态增长
- □ 动态链接

- 6. 下面的()方法有利于程序的动态链接。

 - A. 分段存储管理 B. 分页存储管理

 - C. 可变式分区管理 D. 固定式分区管理
- 引入分段存储管理方式, 主要是为了满足用户和程序员的下述一系列需要:
- □ 方便编程
- □ 信息共享
- □ 信息保护
- □ 动态增长
- □ 动态链接
- A【解析】动态链接是指当程序运行到需要调用某一模块时,再去链接。对于未 使用的模块就可以不必链接。采用段式内存分配方法可以实现这种动态链接。因 此选择A。

- 7. 在分段式存储管理系统中,为了让两个不同的进程共享同一存储段,下列方法中,正确的是()。
 - A. 让进程拥有相同的段表
 - B. 让进程各自的段表项拥有相同的段起始地址和段长度
 - C. 让进程拥有相同的页表
 - D. 不同的进程无法实现共享同一存储段
- B【解析】分段式存储管理系统的段表项包含了段起始地址和段的长度。 两进程共享某一段,就是让进程各自的段表项拥有相同的段起始地址和段长度。 因此选择B。

- 8. 引入分段式存储管理方式,主要是为了更好地满足用户的一系列要求,但不包括()。
 - A. 节约内存 B. 方便编程
 - C. 共享和保护 D. 动态链接和增长

A【解析】引入分段存储管理方式,主要是为了满足用户和程序员的下述一系列需

要:

- □ 方便编程
- □ 信息共享
- □ 信息保护
- □ 动态增长
- □ 动态链接

9. 采用分段式存储管理的系统时,若地址用24位表示,其中8位表示段号,则每 段允许的最大长度是() (默认字长为1 B)。

A. 16 MB B. 64 KB C. 256B D. 4 GB

分段地址中的地址具有如下结构:

	段号			段内地址	
31		16	15		0

B【解析】采用分段式管理的系统中,其逻辑地址分为段号和段内偏移量。本题 的地址共24位,使用了8位表示段号,那么将剩下的16位全部用来表示段内偏移 量就能使每段长度最大,从而每段允许的最大长度为216×1 B=64 KB。因此选择 В.

10. 某段表的内容如表所示。

段号	段首址/KB	段长度(最大长度为64)/KB		
0	120	40		
1	760	30		
2	480	20		
3	370	20		

一逻辑地址为0002 009AH,它对应的物理地址为()。

A. 120 KB+2 B. 480 KB+154 C. 30 KB+154 D. 480 KB+2

B【解析】最大段长度为64 KB,则段内地址位数占16位,前16位表示段号,后16 位表示段内偏移量,

一逻辑地址为0002 009AH,则因此段号为0002H=2,段内偏移量为009AH=154, 对应的物理地址为480 KB+154。因此选择B。

11. 设一段表如表所示。

段号	基地址	段长度/B
0	219	600
1	2300	14
2	90	100
3	1327	580
4	1952	96

那么逻辑地址(2, 88)和(4, 100)对应的物理地址分别是()。

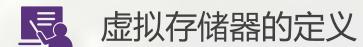
- A. 90,1952 B. 90,2952

- C. 178, 2952 D. 178, 超过段长而产生越界中断
- D【解析】逻辑地址(2, 88)对应的物理地址为90+88=178, 逻辑地址(4, 100) 对应的段号为4,段号为4的段长度为96,100>96,则产生越界中断。 因此选择D。



考点五:

虚拟存储器

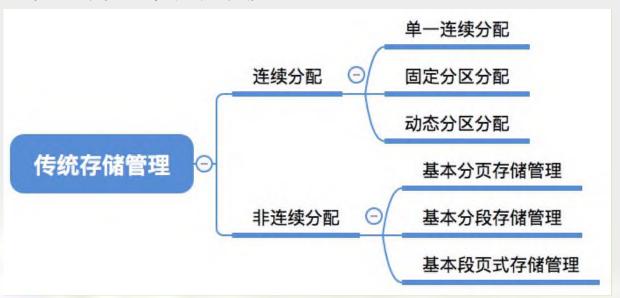




- 页面置换算法
- 抖动和工作集
- 页面调入
- 物理块的分配策略

虚拟存储器的定义

- 0
- 常规存储器管理方式的特征
- □ 一次性: 作业必须一次性全部装入内存后才能开始运行。这会造成两个问题:
- ①作业很大时,不能全部装入内存,导致大作业无法运行;
- ②当大量作业要求运行时,由于内存无法容纳所有作业,因此只有少量作业能运
- 行,导致多道程序并发度下降。





学 常规存储器管理方式的特征

- □ 驻留性: 一旦作业被装入内存, 就会一直驻留在内存中, 直至作业运行结束。
- □ 在一个时间段内,只需要访问作业的一小部分数据即可正常运行,这就导致了 内存中会驻留大量的、暂时用不到的数据,浪费了宝贵的内存资源。





一局部性原理

- □ 时间局部性: 如果程序中的某条指令一旦执行, 则不久以后该指令可能再次执 行;如果某数据被访问过,则不久以后该数据可能再次被访问。
- □ 产生时间局限性的典型原因,是由于在程序中存在着大量的循环操作。

```
#include < stdio.h >
int main() {
  int i;
  while(i++<20) {
     printf("count %d\n",i);
  return 0;
```



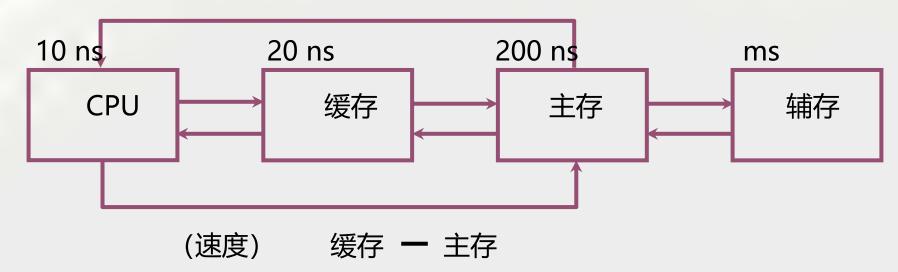
万 局部性原理

- □ 空间局部性: 一旦程序访问了某个存储单元, 在不久之后, 其附近的存储单元 也将被访问,即程序在一段时间内所访问的地址,可能集中在一定的范围之内;
- □典型情况便是程序的顺序执行和数组。

0	1	2	3	4	5	6	7	8



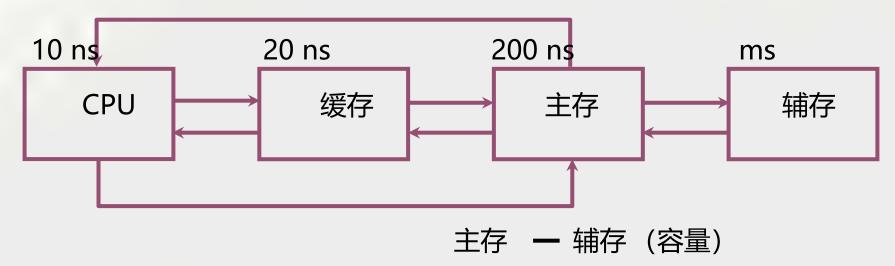
存储器系统的两级架构



□ Cache - 主存层次:缓存-主存层次主要解决CPU 和主存速度不匹配的问题。 由于缓存的速度比主存的速度高,只要将CPU近期要用的信息调入缓存,CPU便可 以直接从缓存中获取信息,从而提高访存速度。通过硬件实现,对用户透明,旨 在解决内存速度与CPU速度不匹配的问题。



存储器系统的两级架构



□ 主存 - 辅存层次: 辅存的速度比主存的速度低,而且不能和CPU直接交换信息,但 它的容量比主存大得多,可以存故大量暂时未用到的信息。当CPU需要用到这些 信息时,再将辅存的内容调入主存,供CPU直接访问,旨在解决内存容量不足的问 题。主存和辅存之间的数据调动是由硬件和操作系统共同完成的。



虚拟存储器的原理

- □ 虚拟存储器, 是指具有请求调入功能和置换功能, 能从逻辑上对内存 容量加以扩充的一种存储器系统。
- □ 其逻辑容量由内存容量和外存容量之和所决定, 其运行速度接近于内 存速度,而每位的成本却又接近于外存。



虚拟存储器的优点

大程序: 可在较小的可用内存中执行较大的用户程序;

大的用户空间: 提供给用户可用的虚拟内存空间通常

大于物理内存

优点 并发:可在内存中容纳更多程序并发执行;

易于开发:不必影响编程时的程序结构

以*CPU*时间和外存空间换取昂贵内存空间,这是操作系统中的资源转换技术

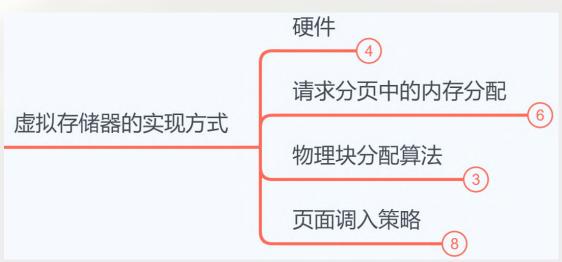


虚拟存储器的特点

	离散性	指在内存分配时采用离散的分配方式,它是虚拟 存储器的实现的基础
特征	多次性	指一个作业被分成多次调入内存运行,即在作业运行时没有必要将其全部装入,只须将当前要运行的那部分程序和数据装入内存即可。多次性是虚拟存储器最重要的特征
15 122	对换性	指允许在作业的运行过程中在内存和外存的对换 区之间换进、换出。
	虚拟性	指能够从逻辑上扩充内存容量,使用户所看到的 内存容量远大于实际内存容量。



虚拟存储器的实现



请求分页存储管理方式

- 分页请求系统
 - (1) 硬件支持
 - □ 请求分页的<mark>页表机制</mark>,它是在纯分页的页表机制上增加若干项而形成的,作 为请求分页的数据结构;
- □ 缺页中断机构,即每当用户程序要访问的页面尚未调入内存时 便产生一缺页中断,以请求OS将所缺的页调入内存;
- □ 地址变换机构, 它同样是在纯分页地址变换机构的基础上发展形成的。
- (2) 实现请求分页的软件。



请求分页中的页表机制

状态位P 外存地址 页号 物理块号 修改位M 访问字段A 缺页中断 0 表示页面 虚拟地址 在系统规 该页内容 该页所占 位: 1表 在内存时数 空间中的 定的时间 存放在辅 内存的块 据未被修改, 示在内存 页号 存中的地 间隔内, 号 1表示被修 中,0表 址,缺页 该页是否 改过。当页 示不在内 时,缺页 被引用过 面被选为淘 存中,为 中断处理 (在页面 0 时会发 汰对象时, 根据它的 淘汰算法 根据此为的 生"缺页" 指点,将 中使用) 中断信号, 取之来确定 所缺的页 是否要将该 请求系统 调入内存 页的内容进 处理

行磁盘回写

操作



请求分页中的页表机制

进程A有5个页面,页号分别是0~4,某个时刻的页表如下所示

进程A

0	
1	
2	
3	
4	

页号	物理块号	状态位	访问字段	修改位	外存地址
0	100	1	20	1	Α
1	-	0	-	-	X
2	1	1	5	0	С
3	-	0	-	-	Т
4	9	1	1	0	N

进程A的页表

从表中可以知道, 0, 2和4在内存中, 1和3不在内存中, 且他们的外存地址是X和T。



请求分页中的页表机制

进程A有5个页面,页号分别是0~4,某个时刻的页表如下所示

进程A	页号	物理	状态	访问	修改位	外存	内存	
0		块号	位	字段		地址	.0	
1	0	100-	1	20	1	Α	1	
2	1	-	0	-	-	/ X	•	1
3	2	1	1	5	0	C	9	
4	3	-	0	-	- /	Т	3	
	4	9	1	1	0	N	100	
A	В	С	•••	N	Т	••	X	



缺页中断机构

①处理过程

- a. 根据当前执行指令中的虚拟地址,形成(页号,页内偏移),用页号查页表, 判断该页是否在内存中
- b. 如该页的缺页中断位为 0, 表示该页面不在内存, 于是产生缺页中断, 让操 作系统的中断处理程序进行中断处理



缺页中断机构

进程A有5个页面,页号分别是0~4,某个时刻的页表如下所示

进程A	页号	物理	状态	访问	修改位	外存	内存
0		块号	位	字段		地址	.0
1	0	100	1	20	1		1
2	1		0	_	-	/ X	•
3	2	1	1	5	0	C	9
4	3	_	0	_	- /	Т	9
4	4	9	1	1	0	N	
			进程A的页				100
	•••						
Α	В	С	•••	N	Т	••	X



缺页中断机构

①处理过程

- c. 中断处理程序查询存储分块表, 寻找一个空闲的内存块; 查询页表, 得到该 页在辅存中的地址, 启动磁盘读信息
- d. 把从磁盘上读出的信息装入到分配的内存块中
- e. 根据分配存储快的信息, 修改页表、存储分块表中相应表目的信息
- f. 由于产生缺页中断的那条指令并未执行, 所以完成所需页面的装入工作后, 应该返回原指令重新执行



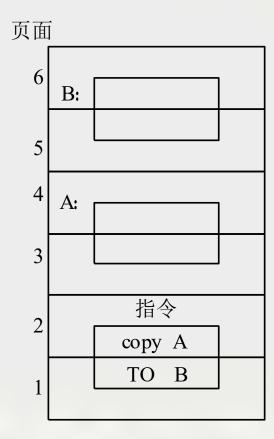
缺页中断机构

- ②缺页中断与一般中断的区别
- a. 缺页中断时在执行一条指令中间时产生的中断,并立即转去处理;一般中断则
- 是在一条指令执行完毕之后, 当发现有中断请求时再去响应和处理
- b. 缺页中断处理执行完毕之后, 仍返回到原指令处重新执行; 一般中断则是返回
- 到下一跳指令去执行



缺页中断机构

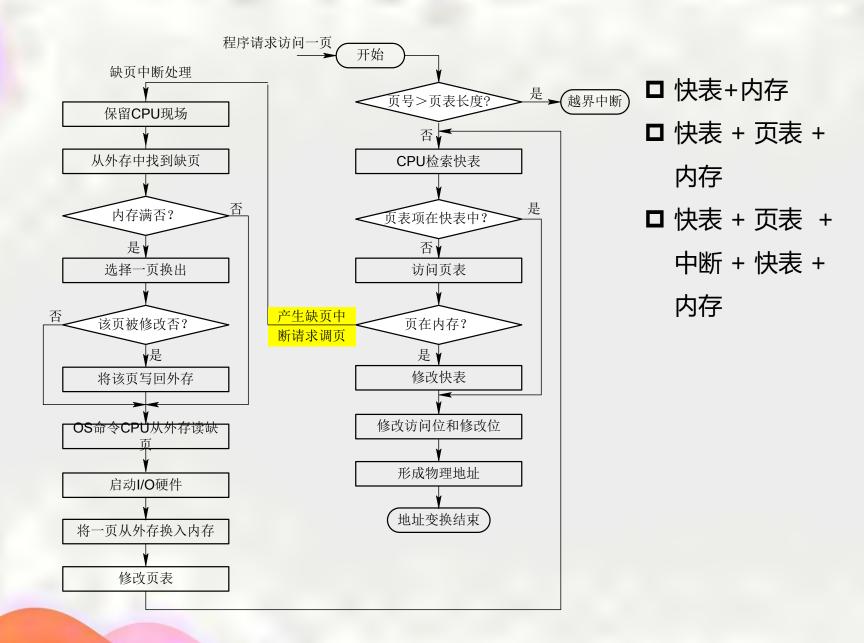
- ②缺页中断与一般中断的区别
- c. 一条指令执行过程中可能会发生多次中断





世 地址变换机构

□ 请求分页系统中的地址变换机构,是在分页系统地址变换机构的基础上,再 为实现虚拟存储器而增加了某些功能而形成的, 如产生和处理缺页中断, 以 及从内存中换出一页的功能等。



页面置换算法



页面置换算法

进程A有5个页面,页号分别是0~4,某个时刻的页表如下所示

1++-		Λ
进	댶	Δ
<u>~</u>	工	, ,

0
1
2
3
4

	页 号	物理	里	状态	访问	修修	り り り り り り り り り り り り り り り り り り り	存		内	存
	号	块-	클	11/	字影		地	址			0
	0	10	0	1	20	1	,	Δ			1
	1	2		0	1	0		X			2
	2	1		1	5	0		C			9
	3	-		0	-	-		Т			••
	4	9		1	1	0		N		1	00
进程A的页表											••
В	3	С		•••	N	•••	Т		•	X	



文 页面置换算法概览





最佳(Optimal)置换算法

□ 最佳置换算法是由 Belady 于 1966 年提出的一种理论上的算法。所选择的被 淘汰页面将是以后永不使用的,或许是在最长(未来)时间内不再被访问的页面。 最佳置换算法通常可保证获得最低的缺页率。



最佳(Optimal)置换算法

□ 假定系统为某进程分配了三个物理块, 并考虑有以下的页面号引用串: 7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2, 0, 1, 7, 0, 1。进程运行时, 其过程如4-20所示,

页面 走向	7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
0	7	7	7	2	2	2	2	2	2	2	2	2	2	2	2	2	2	7	7	7
1		0	0	0	0	0	0	4	4	4	0	0	0	0	0	0	0	0	0	0
2			1	1	1	3	3	3	3	3	3	3	3	1	1	1	1	1	1	1
缺页		√	√	√	Χ	√	Χ	√	Χ	Χ	√	Χ	Χ	√	Χ	Χ	Χ	√	Χ	Χ
置换	×	×	×	√	Χ	√	Χ	√	Χ	Χ	√	Χ	Χ	√	Χ	Χ	Χ	√	Χ	Χ

缺页率 = 9/20

置换率 =6/20



最佳(Optimal)置换算法

□ 由于人们目前还无法预知,一个进程在内存的若干个页面中,哪一个页面是未 来最长时间内不再被访问的,因而该算法是无法实现的,但可以利用该算法去 评价其它算法。

- 先进先出算法 (FIFO)
- □ 先进先出置换算法 (FIFO): 每次选择淘汰的页面是最早进入内存的 页面
- □ 实现方法: 把调入内存的页面根据调入的先后顺序排成一个队列, 需 要换出页面时选择队头页面即可。
- □ 队列的最大长度取决于系统为进程分配了多少个内存块。

先进先出算法 (FIFO)

□ 在一个请求分页系统中,加入一个作业的页面走向为4,3,2,1,4,3,5,4,3,2,1,5,目前它还没有任何页装入内存,当分配给该作业的物理块数目M分别为3和4时,请FIFO页面淘汰算法时,访问过程中所发生的的缺页次数和缺页率,

页面 走向	4	3	2	1	4	3	5	4	3	2	1	5
0	4	4	4	1	1	1	5	5	5	5	5	5
1		3	3	3	4	4	4	4	4	2	2	2
2			2	2	2	3	3	3	3	3	1	1
缺页	√	√	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	×	×	\checkmark	\checkmark	×
置换	×	×	×	√	√	√	√	×	×	√	√	×

缺页率 = 9/12

置换率 = 6/12

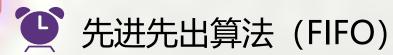
先进先出算法 (FIFO)

□ 在一个请求分页系统中,加入一个作业的页面走向为4,3,2,1,4,3,5,4,3,2,1,5,目前它还没有任何页装入内存,当分配给该作业的物理块数目M分别为3和4时,请FIFO页面淘汰算法时,访问过程中所发生的的缺页次数和缺页率,

页面 走向	4	3	2	1	4	3	5	4	3	2	1	5
0	4	4	4	4	4	4	5	5	5	5	1	1
1		3	3	3	3	3	3	4	4	4	4	5
2			2	2	2	2	2	2	3	3	3	3
3				1	1	1	1	1	1	2	2	2
缺页	√	√	√	√	X	X	√	√	√	√	√	√
置换	×	×	×	×	Χ	Χ	\checkmark	√	\checkmark	\checkmark	\checkmark	√

缺页率 = 10/12

置换率 = 6/12



- Belady奇异现象,是指采用页面置换FIFO算法时,如果对一个进程未分配它 所要求的全部页面,有时就会出现分配的页面数增多,但缺页率反而提高的异 常现象,这是一个违反直觉的现象。
- □ 原因是: 所使用的FIFO算法不够好(不遵循局部性原理)。



世 最近最久未使用置换算法

- □ 最近最久未使用置换算法 (LRU, least recently used): 每次淘汰的页面是 最近最久未使用的页面
- □ 实现方法: 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。



世 最近最久未使用置换算法

□ 在一个请求分页系统中,加入一个作业的页面走向为4,3,2,1,4,3,5,4,3,2,1,5, 目前它还没有任何页装入内存, 当分配给该作业的物理块数目M分别为3和4时, 请LRU页面淘汰算法时,访问过程中所发生的的缺页次数和缺页率,

页面 走向	4	3	2	1	4	3	5	4	3	2	1	5
0	4	4	4	1	1	1	5	5	5	2	2	2
1		3	3	3	4	4	4	4	4	4	1	1
2			2	2	2	3	3	3	3	3	3	5
缺页	√	√	√	\checkmark	\checkmark	√	√	Χ	Χ	√	√	√
置换	×	×	×	√	√	√	√	Χ	Χ	√	√	√

缺页率 = 10/12

置换率 = 7/12



世 最近最久未使用置换算法

□ 在一个请求分页系统中,加入一个作业的页面走向为4,3,2,1,4,3,5,4,3,2,1,5,目 前它还没有任何页装入内存, 当分配给该作业的物理块数目M分别为3和4时, 请LRU页面淘汰算法时,访问过程中所发生的的缺页次数和缺页率,

页面 走向	4	3	2	1	4	3	5	4	3	2	1	5
0	4	4	4	4	4	4	4	4	4	4	4	5
1		3	3	3	3	3	3	3	3	3	3	3
2			2	2	2	2	5	5	5	5	1	1
3				1	1	1	1	1	1	2	2	2
缺页	√	√	√	√	X	X	√	X	X	√	√	\checkmark
置换	×	×	×	×	Χ	Χ	\checkmark	Χ	Χ	\checkmark	\checkmark	\checkmark

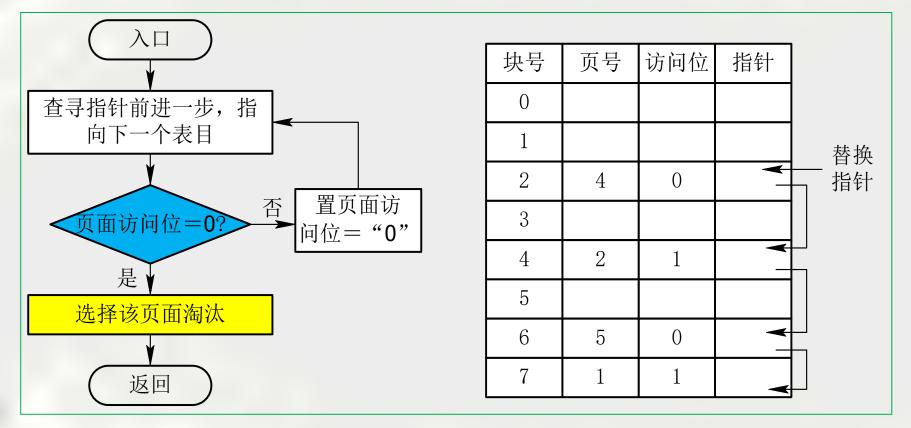
缺页率 = 8/12

置换率 = 4/12

- 简单的Clock置换算法: 被访问页面置1
- □ 当利用简单 Clock 算法时,只需为每页设置一位访问位,再将内存中的所有页面都通过链接指针链接成一个循环队列。
- □ 置换算法在选择一页淘汰时,只需检查页的访问位。如果是 0,就选择该页换出;若为 1,则重新将它置 0,暂不换出,给该页第二次驻留内存的机会,再按照 FIFO 算法检查下一个页面。



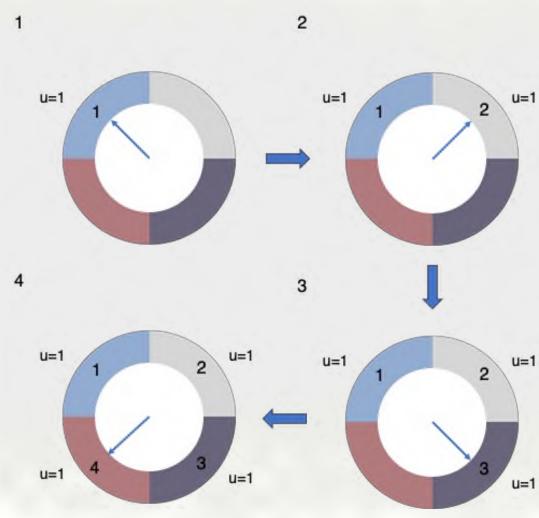
简单的Clock置换算法: 被访问页面置1





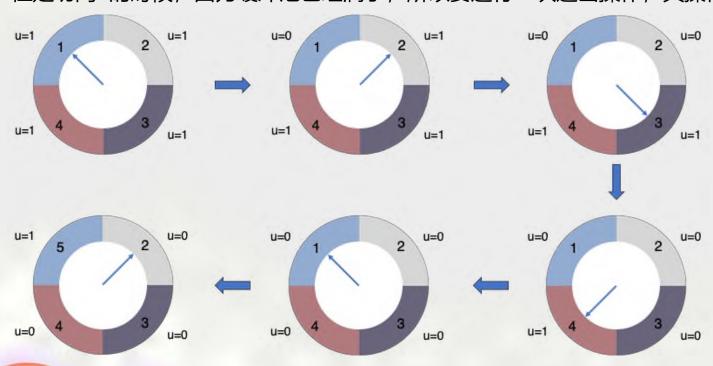
简单的Clock置换算法: 被访问页面置1

我们给每一个页面设置一个标 记位u, u=1表示最近有使用 u=0则表示该页面最近没有被 使用,应该被逐出。按照1-2-3-4的顺序访问页面,则缓冲 池会以这样的一种顺序被填满:



简单的Clock置换算法: 被访问页面置1

注意中间的指针,就像是时钟的指针一样在移动,这样的访问结束后,缓冲池里现在已经被 填满了,此时如果要按照1-5的顺序访问,那么在访问1的时候是可以直接命中缓存返回的, 但是访问5的时候,因为缓冲池已经满了,所以要进行一次逐出操作,其操作示意图如下:





改进型Clock置换算法

在改进型 Clock 算法中,除须考虑页面的使用情况外,还须再增加一个因素—— 置换代价。访问位 A 和修改位 M 可以组合成下面四种类型的页面:

- □ 1类(A=0, M=0):表示该页最近既未被访问,又未被修改,是最佳淘汰页。
- □ 2类(A=0, M=1): 表示该页最近未被访问, 但已被修改, 并不是很好的淘汰页。
- □ 3类(A=1, M=0): 表示该页最近已被访问, 但未被修改, 该页有可能再被访问。
- □ 4类(A=1, M=1): 表示该页最近已被访问且被修改,该页可能再被访问。



改进型Clock置换算法

算法规则:将所有可能被置换的页面排成一个循环队列

- □ 第一轮:从当前位置开始扫描到第一个 (0,0) 的帧用于替换。本轮扫描不修 改任何标志位
- □ 第二轮: 若第一轮扫描失败,则重新扫描,查找第一个(0,1)的帧用于替换。 本轮将所有扫描过的帧访问位设为0
- □ 第三轮: 若第二轮扫描失败,则重新扫描,查找第一个(0,0)的帧用于替换。 本轮扫描不修改任何标志位
- □ 第四轮: 若第三轮扫描失败,则重新扫描,查找第一个(0,1)的帧用于替换。
- □ 由于第二轮已将所有帧的访问位设为0, 因此经过第三轮、第四轮扫描一定会 有一个帧被选中,因此改进型CLOCK置换算法选择一个淘汰页面最多会进行 四轮扫描

页面置换算法总结

	算法规则	优缺点
ОРТ	优先淘汰最长时间内不会被访问的页	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	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	性能很好;但需要硬件 支持,算法开销大
简单 CLOCK	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	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

抖动和工作集



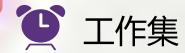
- □ 抖动也称为颠簸,它是指刚刚换出的页面马上又要换入内存,刚刚换入的页面 马上又要换出外存,这种频繁的页面调度行为称为抖动(颠簸)。
- □ 当给进程分配的内存小于所要求的工作集时,由于内存外存之间交换频繁,访问外存时间和输入输出处理时间大大增加,反而造成CPU因等待数据空转,使得整个系统性能大大下降,这就造成了系统抖动。



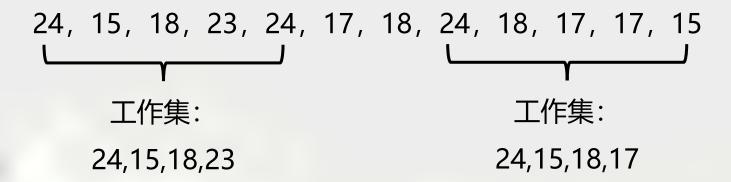
- □ 产生抖动的主要原因是进程频繁访问的页面数目高于可用的物理块数,也就是 分配给进程的物理块不够。
- □ 为进程分配的物理块太少,会使进程发生抖动现象,为进程分配的物理块太多, 又会降低系统整体的并发度,降低某些资源的利用率。抖动将引起严重的系统 性能下降。

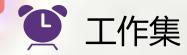


- □ 解决抖动问题:
- □ 1、增加工作集大小;
- □ 2、选择不同的淘汰算法,尽量保持工作集页面在内存中。



- □ 工作集(或驻留集)是指在某段时间间隔内,进程要访问的页面集合。经常被使用的页面需要在工作集中,而长期不被使用的页面要从工作集中被丢弃。为了防止系统出现抖动现象,需要选择合适的工作集大小。
- □ 操作系统会根据 "窗口尺寸"来算出工作集。例:某进程的页面访问序列如下, 窗口尺寸为5,各时刻的工作集





- □ 工作集模型的原理是: 让操作系统跟踪每个进程的工作集,并为进程分配大于其工作集的物理块。
- □ 如果还有空闲物理块,则可以再调一个进程到内存以增加多道程序数。
- □ 如果所有工作集之和增加以至于超过了可用物理块的总数,那么操作系统会暂停一个进程,将其页面调出并且将其物理块分配给其他进程,防止出现抖动现象。

页面调入



量 最小物理块数的确定

- □最小物理块数是指能保证进程正常运行所需的最小物理块数。
- □ 当系统为进程分配的物理块数少于此值时,进程将无法运行。
- □ 进程应获得的最少物理块数与计算机的硬件结构有关, 取决于指令的格 式、功能和寻址方式。



页面调入策略

□ 预调页策略: 启动时, 根据局部性原理, 一次调入若干个相邻的页面可 能比一次调入一个页面更高效。故这种策略主要用于进程的首次调入, 由程序员指出应该先调入哪些部分。



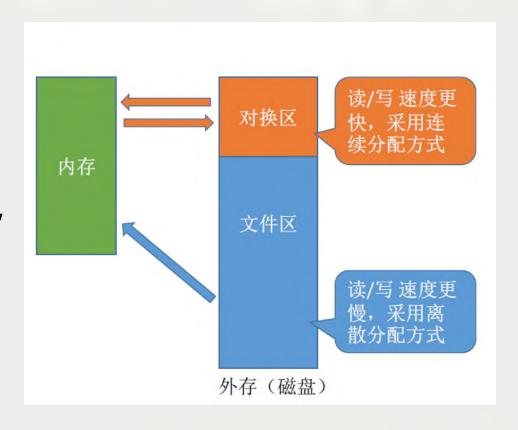
页面调入策略

□ 请求调页策略: 进程在运行期间发现缺页时才将所缺页面调入内存。由 这种策略调入的页面一定会被访问到,但由于每次只能调入一页,而每 次调页都要磁盘I/O操作,因此I/O开销较大。



沙 从何处调入页面

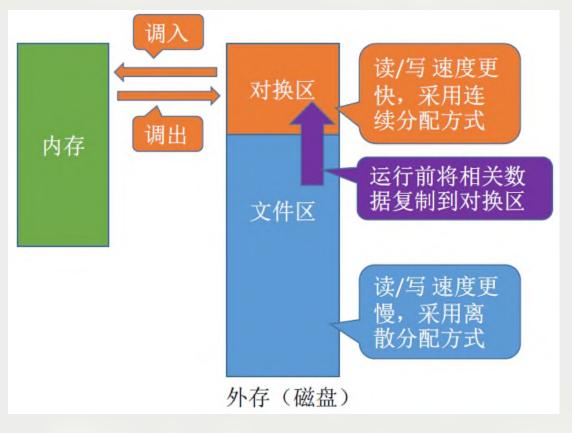
- □ 在请求分页系统中的外存分为两 部分:用于存放文件的文件区和 用于存放对换页面的对换区。
- □ 由于**对换区**是采用**连续分配**方式, 而文件区是采用离散分配方式, 故对换区的磁盘I/O速度比文件 区的高。





少 从何处调入页面

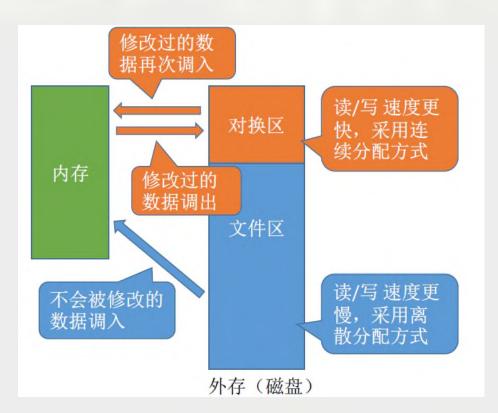
口 系统拥有足够的对换区空间, 这时可以全部从对换区调入 所需页面,以提高调页速度。 为此, 在进程运行前, 便须 将与该进程有关的文件,从 文件区拷贝到对换区。





沙 从何处调入页面

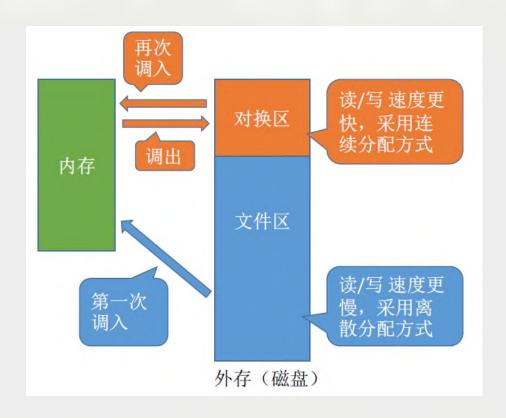
口 系统缺少足够的对换区空间,这时凡 是不会被修改的文件,都直接从文件 区调入; 而当换出这些页面时, 由于 它们未被修改而不必再将它们换出, 以后再调入时,仍从文件区直接调入。 但对于那些可能被修改的部分,在将 它们换出时,便须调到对换区,以后 需要时,再从对换区调入。





沙 从何处调入页面

□ UNIX方式。由于与进程有关的文件 都放在文件区, 故凡是未运行过的页 面,都应从文件区调入。而对于曾经 运行过但又被换出的页面,由于是被 放在对换区,因此在下次调入时,应 从对换区调入。由于UNIX系统允许 页面共享,因此, 某进程所请求的页 面有可能已被其它进程调入内存,此 时也就无须再从对换区调入。





变 页面调入过程

- 口 每当程序所要访问的页面未在内存时,便向CPU发出一缺页中断,中断处理 程序首先保留CPU环境,分析中断原因后, 转入缺页中断处理程序。
- 口 该程序通过查找页表,得到该页在外存的物理块后,如果此时内存能容纳新 页,则启动磁盘I/O将所缺之页调入内存,然后修改页表。



亚 页面调入过程

- 口 如果内存已满,则须先按照某种置换算法从内存中选出一页准备换出;
 - 口 如果该页未被修改过,可不必将该页写回磁盘;
 - 口 如果此页已被修改,则必须将它写回磁盘,然后再把所缺的页调入内 存, 并修改页表中的相应表项,置其存在位为"1",并将此页表项 写入快表中。
- 口 在缺页调入内存后,利用修改后的页表, 去形成所要访问数据的物理地址, 再去访问内存数据。

物理块的分配策略



量 最小物理块数的确定

- □最小物理块数是指能保证进程正常运行所需的最小物理块数。
- □ 当系统为进程分配的物理块数少于此值时,进程将无法运行。
- □ 进程应获得的最少物理块数与计算机的硬件结构有关, 取决于指令的格 式、功能和寻址方式。



物理块的分配策略

- □ 固定分配:操作系统为每个进程分配—组固定数目的物理块,在进程运行期间 不再改变。即, 驻留集大小不变
- □ 可变分配: 先为每个进程分配一定数目的物理块, 在进程运行期间, 可根据情 况做适当的增加或减少。即, 驻留集大小可变
- □ 局部置换:发生缺页时只能选进程自己的物理块进行置换。
- □ 全局置换:可以将操作系统保留的空闲物理块分配给缺页进程,也可以将别的 进程持有的物理块置换到外存, 再分配给缺页进程。



物理块的分配策略

□ 在请求分页系统中,可采取两种内存分配策略,即固定和可变分配策 略。在进行置换时,也可采取两种策略,即全局置换和局部置换。于 是可组合出以下三种适用的策略。

	全局置换	局部置换
固定分配	×	√
可变分配	V	\checkmark

全局置换意味着一个 进程拥有的物理块数 量必然会改变,因此 不可能是固定分配



物理块的分配策略

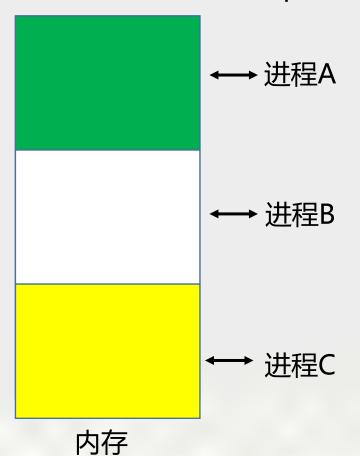
- □ 固定分配局部置换(Fixed Allocation, Local Replacement) (固定分 配,置换自身进程页)
- 系统为每个进程分配一定数量的物理块,在整个运行期间都不改变。 若进程在运行中发生缺页,则只能从该进程在内存中的页面中选出一 页换出, 然后再调入需要的页面。
- ▶ 这种策略的缺点是: 很难在刚开始就确定应为每个进程分配多少个物 理块才算合理。



物理块的分配策略

口 固定分配局部置换(Fixed Allocation, Local Replacement) (固定分

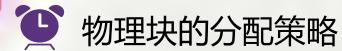
配,置换自身进程页)





物理块的分配策略

- □ 可变分配全局置换(Variable Allocation, Global Replacement) (动态分配, 物理块用完时可置换其他进程页)
- 刚开始会为每个进程分配一定数量的物理块。操作系统会保持一个空闲物理块队列。当某进程发生缺页时,从空闲物理块中取出一块分配给该进程;若已无空闲物理块,则可选择一个未锁定的页面换出外存,再将该物理块分配给缺页的进程。
- 采用这种策略时,只要某进程发生缺页,都将获得新的物理块,仅当空闲物理块用完时,系统才选择一个未锁定的页面调出。
- 被选择调出的页可能是系统中任何一个进程中的页,因此这个被选中的进程拥有的物理块会减少,缺页率会增加



□ 可变分配全局置换(Variable Allocation, Global Replacement) (动态分配,物理块用完时可置换其他进程页)





物理块的分配策略

- □ 可变分配局部置换(Variable Allocation, Local Replacemen (置换自 身进程页,频繁缺页时动态分配)
- > 可变分配局部: 刚开始会为每个进程分配一定数量的物理块。当某进程发生缺页 时,只允许从该进程自己的物理块中选出一个进行换出外存。
- > 如果进程在运行中频繁地缺页,系统会为该进程多分配几个物理块,直至该进程 缺页率趋势适当程度; 反之, 如果进程在运行中缺页率特别低, 则可适当减少分 配给该进程的物理块



物理块的分配策略

ロ 可变分配局部置换(Variable Allocation, Local Replacemen (置换自 身进程页,频繁缺页时动态分配)

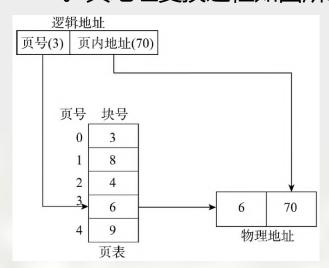


【政哥点拨】

1. 假定某页式管理系统中,主存为128KB,分成32块,块号为0,1,2,3,...,31。 某作业有5块,其页号为0,1,2,3,4,被分别装入主存的3,8,4,6,9块中。有一逻辑 地址为[3,70],则相应的物理地址为(方括号中的第一个元素为页号,第二个元 素为页内地址,按十进制计算)()。

A. 14646 B. 24646 C. 24576 D. 34576

B【解析】块大小为128 KB/32=4 KB, 因为块与页面大小相等, 所以每页为4KB。第3页被装入主存第6块中, 故逻辑地址[3, 70]对应的物理地址为4KB×6+70=24576+70=24646。其地址变换过程如图所示。



2. 系统为某进程分配了4个页框,该进程已访问的页号序列为2,0,2,9,3,4,2,8,2,4,8,4,5。若进程要访问的下一页的页号为7,依据LRU算法,淘汰页的页号是()。

A. 2 B. 3 C. 4 D. 8

A【解析】最近最久未使用(LRU)算法每次执行页面置换时会换出最近最久未使用过的页面。具体的页面置换情况如表所示。

访问页面	2	0	2	9	3	4	2	8	2	4	8	4	5
物理块1	2	2	2	2	2	2	2	2	2	2	2	2	2
物理块2		0	0	0	0	4	4	4	4	4	4	4	4
物理块3				9	9	9	9	8	8	8	8	8	8
物理块4					3	3	3	3	3	3	3	3	5
缺页否	√	√		√	√	√		√					√

当要访问的下一个页号为7时,根据LRU算法,应淘汰的页号为2。

- 3. 在一个请求分页存储管理系统中,一个作业的页面走向为4,3,2,1,4,3,5,4,3,2,1,5,当分配给作业的物理块数分别为3和4时,试计算采用下述页面淘汰算法时的缺页率(假设开始执行时主存中没有页面),并比较结果。
- (1)最佳置换算法。
- (2)先进先出置换算法。
- (3)最近最久未使用算法。

【解析】(1)根据页面走向,物理块数为3时,使用最佳置换算法,页面置换情况如表所所示

走向	4	3	2	1	4	3	5	4	3	2	1	5
块1	4	4	4	4	4	4	4	4	4	2	2	2
块2		3	3	3	3	3	3	3	3	3	1	1
块3			2	1	1	1	5	5	5	5	5	5
缺页	√	√	√	√			√			√	√	

缺页率为7/12。

- 3. 在一个请求分页存储管理系统中,一个作业的页面走向为4,3,2,1,4,3,5,4,3,2,1,5,当分配给作业的物理块数分别为3和4时,试计算采用下述页面淘汰算法时的缺页率(假设开始执行时主存中没有页面),并比较结果。
- (1)最佳置换算法。
- (2)先进先出置换算法。
- (3)最近最久未使用算法。

【解析】(1)根据页面走向,物理块数为4时,使用最佳置换算法,页面置换情况如表所所示

走向	4	3	2	1	4	3	5	4	3	2	1	5
块1	4	4	4	4	4	4	4	4	4	4	1	1
块2		3	3	3	3	3	3	3	3	3	3	3
块3			2	2	2	2	2	2	2	2	2	2
块4				1	1	1	5	5	5	5	5	5
缺页	√	√	√	√			√				√	

缺页率为6/12。

由上述结果可以看出,增加分配作业的内存块数可以降低缺页率。

- 3. 在一个请求分页存储管理系统中,一个作业的页面走向为4,3,2,1,4,3,5,4,3,2,1,5,当分配给作业的物理块数分别为3和4时,试计算采用下述页面淘汰算法时的缺页率(假设开始执行时主存中没有页面),并比较结果。
- (1)最佳置换算法。
- (2)先进先出置换算法。
- (3)最近最久未使用算法。

【解析】(1)根据页面走向,物理块数为3时,使用先进先出置换算法,页面置换情况如表所所示

走向	4	3	2	1	4	3	5	4	3	2	1	5
块1	4	4	4	1	1	1	5	5	5	5	5	5
块2		3	3	3	4	4	4	4	4	2	2	2
块3			2	2	2	3	3	3	3	3	1	1
缺页	√	√	√	√	√	√	√			√	√	

缺页率为9/12。

- 3. 在一个请求分页存储管理系统中,一个作业的页面走向为4,3,2,1,4,3,5,4,3,2,1,5,当分配给作业的物理块数分别为3和4时,试计算采用下述页面淘汰算法时的缺页率(假设开始执行时主存中没有页面),并比较结果。
- (1)最佳置换算法。
- (2)先进先出置换算法。
- (3)最近最久未使用算法。

【解析】(1)根据页面走向,物理块数为4时,使用先进先出置换算法,页面置换情况如表所所示

走向	4	3	2	1	4	3	5	4	3	2	1	5
块1	4	4	4	4	4	4	5	5	5	5	1	1
块2		3	3	3	3	3	3	4	4	4	4	5
块3			2	2	2	2	2	2	3	3	3	3
块4				1	1	1	1	1	1	2	2	2
缺页	√	√	√	√			√	√	√	√	√	√

缺页率为10/12。 由上述结果可以看出,对先进先出算法而言,增加分配作业的内存块数反而使缺页率上升,即出现Belady现象。

- 3. 在一个请求分页存储管理系统中,一个作业的页面走向为4,3,2,1,4,3,5,4,3,2,1,5,当分配给作业的物理块数分别为3和4时,试计算采用下述页面淘汰算法时的缺页率(假设开始执行时主存中没有页面),并比较结果。
- (1)最佳置换算法。
- (2)先进先出置换算法。
- (3)最近最久未使用算法。

【解析】(1)根据页面走向,物理块数为3时,使用最近最久未使用算法,页面置换情况如表所所示

走向	4	3	2	1	4	3	5	4	3	2	1	5
块1	4	4	4	1	1	1	5	5	5	2	2	2
块2		3	3	3	4	4	4	4	4	4	1	1
块3			2	2	2	3	3	3	3	3	3	5
缺页	√	√	√	√	√	√	√			√	√	√

缺页率为10/12。

- 3. 在一个请求分页存储管理系统中,一个作业的页面走向为4,3,2,1,4,3,5,4,3,2,1,5,当分配给作业的物理块数分别为3和4时,试计算采用下述页面淘汰算法时的缺页率(假设开始执行时主存中没有页面),并比较结果。
- (1)最佳置换算法。
- (2)先进先出置换算法。
- (3)最近最久未使用算法。

【解析】(1)根据页面走向,物理块数为4时,使用最近最久未使用算法,页面置换情况如表所所示

走向	4	3	2	1	4	3	5	4	3	2	1	5
块1	4	4	4	4	4	4	4	4	4	4	4	5
块2		3	3	3	3	3	3	3	3	3	3	3
块3			2	2	2	2	5	5	5	5	1	1
块4				1	1	1	1	1	1	2	2	2
缺页	√	√	√	√			√			√	√	√

缺页率为8/12。

由上述结果可以看出,增加分配作业的内存块数可以降低缺页率。

4. 在请求分页管理系统中, 假设某进程的页表内容如表所示。

页号	页框(Page Frame)号	有效位 (存在位)
0	101H	1
1	_	0
2	254H	1

页面大小为4KB,一次内存的访问时间是100ns,一次快表(TLB)的访问时间是10ns,处理一次缺页的平均时间为10^8ns(已含更新TLB和页表的时间),进程的驻留集大小固定为2,采用最近最久未使用(LRU)置换算法和局部淘汰策略。

- ①TLB初始为空;
- ②地址转换时先访问TLB,若TLB未命中,再访问页表(忽略访问页表后的TLB更新时间);
- ③有效位为0表示页面不在内存,产生缺页中断,缺页中断处理后,返回到产生缺页中断的指令处重新执行。

设虚拟地址访问序列2362H, 1565H, 25A5H, 请问:

(1)依次访问上述三个虚拟地址, 各需多少时间? 给出计算过程。

【解析】(1)根据分页管理的工作原理,应先考虑页面大小,以便将页号和页内位移分解出来。页面大小为4 KB,即2¹²,得到页内位移占虚拟地址的低12位,页号占剩余高位。可得三个虚拟地址的页号P如下(十六进制的1位数字转换成二进制的4位数字,因此十六进制的低三位正好为页内位移,最高位为页号):

- □ 快表+内存
- □ 快表 + 页表 + 内存
- □ 快表 + 页表 +中断 + 快表 +内存

- ①TLB初始为空;
- ②地址转换时先访问TLB,若TLB未命中,再访问页表(忽略访问页表后的TLB更新时间);
- ③有效位为0表示页面不在内存,产生缺页中断,缺页中断处理后,返回到产生缺页中断的指令处重新执行。

设虚拟地址访问序列2362H, 1565H, 25A5H, 请问:

- (1)依次访问上述三个虚拟地址, 各需多少时间? 给出计算过程。
- (1)依次访问上述三个虚拟地址, 各需多少时间? 给出计算过程。

【解析】(1) 2362H: P=2, 访问快表10ns, 因初始为空, 访问页表100ns得到页框号, 合成物理地址后访问主存100ns, 共计10ns+100ns=210(ns)。

- □ 快表+内存
- □ 快表 + 页表 + 内存
- □ 快表 + 页表 +中断 + 快表 +内存

- ①TLB初始为空;
- ②地址转换时先访问TLB,若TLB未命中,再访问页表(忽略访问页表后的TLB更新时间);
- ③有效位为0表示页面不在内存,产生缺页中断,缺页中断处理后,返回到产生缺页中断的指令处重新执行。

设虚拟地址访问序列2362H, 1565H, 25A5H, 请问:

(1)依次访问上述三个虚拟地址, 各需多少时间? 给出计算过程。

【解析】(1) 1565H: P=1, 访问快表10ns, 落空, 访问页表100ns落空, 进行缺页中断处理108ns, 访问快表10ns, 合成物理地址后访问主存100ns, 共计10ns+100ns+108ns+10ns+100ns=100000220(ns)。

- □ 快表+内存
- □ 快表 + 页表 + 内存
- □ 快表 + 页表 +中断 + 快表 +内存

- ①TLB初始为空;
- ②地址转换时先访问TLB,若TLB未命中,再访问页表(忽略访问页表后的TLB更新时间);
- ③有效位为0表示页面不在内存,产生缺页中断,缺页中断处理后,返回到产生缺页中断的指令处重新执行。

设虚拟地址访问序列2362H, 1565H, 25A5H, 请问:

(1)依次访问上述三个虚拟地址, 各需多少时间? 给出计算过程。

【解析】(1) 25A5H: P=2, 访问快表, 因第一次访问已将该页号放入快表, 因此花费10ns便可合成物理地址, 访问主存100ns, 共计10ns+100ns=110(ns)。

- □ 快表+内存
- □ 快表 + 页表 + 内存
- □ 快表 + 页表 +中断 + 快表 +内存

- ①TLB初始为空;
- ②地址转换时先访问TLB, 若TLB未命中, 再访问页表(忽略访问页表后的TLB更新时间);
- ③有效位为0表示页面不在内存,产生缺页中断,缺页中断处理后,返回到产生缺页中断的指令处重新执行。

设虚拟地址访问序列2362H, 1565H, 25A5H, 请问:

(2) 基于上述访问序列, 虚地址1565H的物理地址是多少? 请说明理由。

【解析】(2)当访问虚拟地址1565H时,产生缺页中断,合法驻留集大小为2,必须从页表中淘汰一个页面,根据题目的置换算法,应淘汰0号页面,因此1565H的对应页框号为101H。由此可得1565H的物理地址为101565H。

页号	页框(Page Frame)号	有效位 (存在位)
0	_	0
1	101H	1
2	254H	1

【牛刀小试】

- 1. 【南京理工大学 2013,广东工业大学 2014】在请求页式存储管理中,若所需页面不在内存中,则会引起()。
 - A. 输入输出中断 B. 时钟中断 C. 越界中断 D. 缺页中断
- D【解析】在请求页式存储管理中,若所需页面不在内存中,则会引起缺页中断。 因此选择D。

- 2. 【广东工业大学 2017】请求页式管理中,缺页中断率与进程所分得的内存页面数、()和进程页面流的走向等因素有关。
 - A. 页表的位置 B. 置换算法
 - C. 外存管理算法 D. 进程调度算法
- B【解析】请求页式管理中,缺页中断率与进程所分得的内存页面数、置换算法和进程页面流的走向等因素有关。因此选择B。

- 3. 【北京交通大学 2016】关于页表 (进程)的页表项,基本分页存储管理方式和请求分页存储管理方式均须设立的字段为()。
 - A. 状态位 B. 访问字段
 - C. 修改位 D. 物理块号
- D【解析】基本分页存储管理方式和请求分页存储管理方式均须设立的字段为物理块号。请求页表的管理方式比基本页表方式的页表要多加四个字段:
- (1)状态位,是否已经调入内存;
- (2)访问字段,可记录最近被访问过几次,或记录上次访问的时间,供置换算法选择换出页面时进行参考;
- (3)修改位,页面调入内存后是否被修改过;
- (4)外存地址,页面在外存中的存放位置。因此选择D。

- 4. 【北京交通大学 2018】对于请求分段内存管理系统而言,进程段表项中用来描述对应分段是否调入内存的是()。
 - A. 修改位 B. 存在位
 - C. 增补位 D. 访问字段

B【解析】

修改位:表示该页在调入内存后是否被修改过,供置换页面时参考;

存在位: 本段是否已调入内存, 供程序访问参考;

增补位:表示本段在运行过程中是否动态增长;

访问字段:用于记录该字段被访问的频繁程度。

- 5. 测得某个采用按需调页(Demand-paging)策略的计算机系统部分状态数据: CPU利用率为20%,用于对换空间的硬盘利用率为97.7%,其他设备的利用率为5%。由此断定系统出现异常。此种情况下()能提高利用率。
 - A. 安装一个更快的硬盘
 - B. 通过扩大硬盘容量增加对换空间
 - C. 增加运行进程数
 - D. 加内存条来增加物理空间容量
- D【解析】CPU利用率低,对换空间的硬盘利用率高,说明程序进出内存频率高,很可能是内存抖动,此时需要增大物理内存空间,或者降低运行进程数。因此选择D。

6. 在采用请求分页式存储管理的系统中,地址变换过程可能会因为下列()而产生中断。

Ⅰ. 地址越界 Ⅱ. 缺页

Ⅲ.访问权限错误 IV. 存取控制

A. 仅I和I B. 仅I和II

C. 仅I、I和II D. 仅I、II和IV

C【解析】在地址变换过程中,可能会因为缺页、操作保护和越界保护而产生中断。

- □ 当访问的页内地址超过页长度时,就发生了地址越界,
- □ 当访问的页面不在内存中时,就会产生缺页中断,
- □ 当访问权限错误是在执行的操作与页表中保护位(比如读写位、用户/系统属性位等)不一致时发生,比如对一些代码页执行了写操作,而这些代码页是不允许写操作的。因此选择C。

- 7. 在请求页式存储管理中,页面的大小与可能产生的缺页中断次数()。
 - A. 成正相关 B. 成反相关
 - C. 无关 D. 成固定比例
- B【解析】页面越大,其包含的指令和数据越多,页面数量就越少,每次新页面加载后到下次发生缺页的时间就越长,也就是缺页中断的次数越少。因此选择B。

8. 假设一个请求分页系统具有一个平均访问和传输时间为20ms的分页硬盘,为了提高性能,加入了快表,多数活动页表项都可以存在其中。如果页表存放在内存中,内存访问时间是1µs,检索快表的时间为0.2µs,若快表的命中率是80%,未命中快表访问中的50%会导致页错误,则内存的有效存取时间是()。

- A. 1001.4μs B. 1401.6μs
- C. 2001.4µs D. 2401.6µs

C【解析】80%的访问在相关存储器中,耗时1.2μs (快表+内存); 10%的访问需在页表和内存中进行,两次访问内存总共耗时2μs (页表+内存); 10%的访问在页表中产生页错误(1μs),经调页系统工作后(20 ms)再访问内存(1μs),共耗时20002.2μs (页表 +中断 + 快表+ 页表); 有效访问时间为0.8×1.2μs+0.1×2μs+0.1×20002.2μs=2001.38(μs)。因此选择C。

- 9. 下列关于请求分页存储管理说法中,不正确的是()。
 - A. 程序空间页的大小与计算机物理块的大小总是一致的
 - B. 页地址变换机构必须由相应的硬件支持
 - C. 将用户地址空间分为页号和页内偏移,用户是感觉不到的
 - D. 在请求调页的系统中,用户程序必须全部装入主存
- D【解析】程序空间页的大小与计算机物理块的大小总是一致的, A正确;

页地址变换机构必须由相应的硬件支持, B正确;

将用户地址空间分为页号和页内偏移,用户是感觉不到的,C正确;

在请求调页的系统中,用户程序不需要全部装入主存,D错误。因此选择D。

- 10. 下列关于请求分段存储管理的叙述中,正确的是()。
 - A. 分段的尺寸受内存空间的限制, 且作业总的尺寸也受到内存空间的限制
 - B. 分段的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
 - C. 分段的尺寸不受内存空间的限制,且作业总的尺寸也不受内存空间的限制
 - D. 分段的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
- B【解析】分段对应的是内存空间管理的一种方式,采用段内地址+基地址来确定 访存地址,分段尺寸最大为实际内存大小。在动态链接时先将主程序所对应的目标程序装入内存并启动运行,运行过程中需要调用某段时才将该段内存合并进行链接。而作业的大小不受内存大小限制,由虚拟存储器解决空间不够问题,允许作业装入的时候只装入一部分,另一部分放在磁盘上,当需要的时候再装入主存,这样一来,在一个小的主存空间就可以运行一个比它大的作业。同时,用户编程的时候也摆脱了一定要编写小于主存容量的作业的限制。因此选择B。

- 11. 在请求调页系统中,若逻辑地址中的页号超过页表控制寄存器中的页表长度,则会引起()。
 - A. 输入/输出中断 B. 时钟中断
 - C. 越界中断 D. 缺页中断
- C【解析】在请求调页系统中,若逻辑地址中的页号超过页表控制寄存器中的页表长度,则会引起越界中断。因此选择C。

- 12. 【中国计量大学 2019】在基本分页存储管理中,若采用最优页面置换算法 OPT,则当进程分配到的物理块数增加时,那么缺页中断的次数将()。
 - A. 一定会减少 B. 一定不会增加
 - C. 无影响 D. 可能减少也可能增加

A【解析】最优页面置换算法中的每个页面都可以将该页面首次被访问前所要执行的指令数作为标记。最优页面置换算法规定应该置换标记最大的页面或者在最长时间内不再被访问的页面。每次选择淘汰的页面将永不使用,这样可以保证最低的缺页率。当内存物理块增加时,缺页中断的次数会减少。因此选择A。

- 13. 在虚拟分页存储管理系统中,若进程访问的页面不在主存中,且主存中没有可用的空闲帧时,系统正确的处理顺序为()。
 - A. 决定淘汰页→页面调出→缺页中断→页面调入
 - B. 决定淘汰页→页面调入→缺页中断→页面调出
 - C. 缺页中断→决定淘汰页→页面调出→页面调入
 - D. 缺页中断→决定淘汰页→页面调入→页面调出
- C【解析】虚拟分页存储管理的步骤:在程序运行中发现所需要的页不在物理内存时,首先发出缺页中断;然后根据算法决定淘汰哪页;接着把物理内存中的淘汰页存储到外存,称为页面调出;最后把需要的内容从外存调入物理内存指定页,称为页面调入。因此选择C。

- 14. 【广东工业大学 2014】在请求分页系统中, LRU算法是指()。
 - A. 最早进入内存的页先淘汰
 - B. 近期最长时间以来没被访问的页先淘汰
 - C. 近期被访问次数最少的页先淘汰
 - D. 以后再也不用的页先淘汰
- B【解析】最近最久未使用(LRU)算法是根据页面调入内存后的使用情况,将最近最久未使用的页面淘汰。因此选择B。

15. 【中国计量大学 2017】在虚拟存储系统中,若进程在内存中占3块(开始都为空),采用先进先出的页面淘汰算法,当执行访问页号顺序为1,2,3,4,1,2,5,1,2,3,4,5,6时,将产生()次缺页中断。

A. 7 B. 8 C. 9 D. 10

D【解析】根据表可得, 共发生了10次缺页中断。因此选择D。

序列	1	2	3	4	1	2	5	1	2	3	4	5	6
	1	1	1	4	4	4	5	5	5	5	5	5	5
3个 内存块		2	2	2	1	1	1	1	1	3	3	3	3
1313-50			3	3	3	2	2	2	2	2	4	4	4
缺页	*	*	*	*	*	*	*			*	*		*

- 16. 【中国计量大学 2020】在请求分页存储管理中,若采用先进先出的FIFO页面置换算法,则当进程分配到的物理块数增加时,那么缺页中断的次数将()。
 - A一定会减少 B. 一定不会增加
 - C. 无影响 D. 可能减少也有可能增加
- D【解析】根据缺页率增减进程的物理块数:若频繁缺页中断,则系统再为进程分配若干物理块;若缺页率特别低,则适当减少分配给该进程的物理块。缺页率随着所分配物理块数的增加明显地减少,当物理块数超过某个数目时,再为进程增加一物理块,对缺页率的改善并不明显。可见,此时已无必要再为它分配更多的物理块。反之,当为某进程所分配的物理块数低于某个数目时,每减少一块,对缺页率的影响都变得十分明显,此时又应为该进程分配更多的物理块。因此选择D。

- 17. 【南京理工大学 2013】下面的页面淘汰算法会产生Belady异常现象的是()。
 - A. 先进先出页面淘汰算法(FIFO)
 - B. 最近最少使用页面淘汰算法(LRU)
 - C. 最不经常使用页面淘汰算法(LFU)
 - D. 最佳页面淘汰算法(OPT)

A【解析】先进先出页面淘汰算法(FIFO)会产生Belady异常现象。因此选择A。

【经典总结】先进先出页面淘汰算法(FIFO)会产生Belady异常现象。操作系统维持一个所有当前在内存中的页的链表,最老的页在表头,最新来的页在表尾。当发生页面故障时淘汰表头的页并把新调入的页加到表尾。由于FIFO只是简单地淘汰最老的页面,而老页面不一定是最没用的,所以单纯形式的FIFO很少使用。

18. 【南京理工大学 2017】系统采用页式虚拟存储管理和固定分配局部置换策略,页框大小为512 B。某个进程中有如下代码段:

```
1  int a[128][128];
2  for(int i=0; i<128; i++)
3  for(int j=0; j<128; j++)
4  a[i][j]=0;</pre>
```

假设系统为每个int型数据分配4B空间,该段代码已在内存,但要处理的数据不在内存。系统为该进程分配的数据区只有1个页框,则执行上述代码会发生()次缺页中断(假设页面置换时不置换代码段)。

A.1 B. 2 C. 128 D. 16384

C【解析】页框大小为512 B,每个int型数据分配4B空间,512 B/4 B=128,则一个页框可以装128个int型数据,数据区只有1个页框,因此会发生128次缺页中断。因此选择C。

19. 【华东师范大学 2015】在请求分页系统中,一个进程分配到m个物理块,这些物理块初始时全空,页面引用串长度为p,其中包含了n个不同的页号,则无论采用何种页面置换算法,该进程运行过程中的缺页次数k满足()。

A. m < k < p B. $m \le k \le p$ C. n < k < p D. $n \le k \le p$

D【解析】无论采用何种页面置换算法,每种页面第一次访问时不可能在内存中, 必然发生缺页,所以缺页次数大于等于n,则n≤k≤p。因此选择D。

- 20. 下面关于请求页式系统的页面调度算法中, 说法错误的是()。
- A. 一个好的页面调度算法应减少和避免抖动现象
- B. FIFO算法实现简单,选择最先进入主存储器的页面调出
- C. LRU算法基于局部性原理,首先调出最近一段时间内最长时间未被访问过的页面
- D. CLOCK算法首先调出一段时间内被访问次数多的页面
- D【解析】CLOCK算法调出未使用过的页面。因此选择D。

【经典总结】一个好的页面调度算法应减少和避免抖动现象; FIFO算法实现简单, 选择最先进入主存储器的页面调出; LRU算法基于局部性原理, 首先调出最近一段时间内最长时间未被访问过的页面; CLOCK算法调出未使用过的页面。

- 21.请求分页存储管理的主要特点是()。
 - A. 消除了页内零头 B. 扩充了内存
 - C. 便于动态链接 D. 便于信息共享
- B【解析】请求分页存储管理就是为了解决内存容量不足而使用的方法,它基于局部性原理实现了以时间换取空间的目的。它的主要特点自然是间接扩充了内存。因此选择B。

- 22. 在请求分页存储管理的页表中增加了若干项信息,其中修改位和访问位供()参考。
 - A. 分配页面 A. 调入页面
 - C. 置换算法 D. 程序访问
- C【解析】当需要置换页面时,置换算法根据修改位和访问位选择调出内存的页面。因此选择C。

- 23. 产生内存抖动的主要原因是()。
 - A. 内存空间太小 B. CPU运行速度太慢
 - C. CPU调度算法不合理 D. 页面置换算法不合理
 - D【解析】内存抖动是指频繁地引起主存页面淘汰后又立即调入,调入后又很快淘汰的现象。这是由页面置换算法不合理引起的一种现象,是页面置换算法应尽量避免的。因此选择D。

- 24. 用软件方式实现LRU算法开销大的主要原因是()。

 - A. 需要硬件的特殊支持 B. 需要特殊的中断处理程序
 - C. 需要在页表中标明特殊的页类型 D. 需要对所有的页进行排序
- D【解析】LRU算法需要对所有的页最近一次被访问的时间进行记录,查找时间最 久的进行替换,这涉及排序,对置换算法而言,开销太大。因此选择D。

【大显身手】

1.【南京理工大学 2017】请求分页管理系统中,页面大小为1KB,采用固定分配局部置换策略。访存一次时间为50ns,访快表一次的时间为10ns,处理一次缺页的平均时间为50ms(已包括更新TLB和页表的时间)。设快表初始为空,快表未命中时,忽略快表的更新时间。系统为某进程分配3个页框,其页表如表所示。设该进程有逻辑地址访问序列256H,D65H,AB7H,456H,6B7H,若采用CLOCK置换算法(查询被置换页的指针初始在0号页,查找顺序按表由上至下)。

页号	页框号	访问位
0	6	1
1	7	0
2	4	0

(1)请问遵循上述地址访问序列,访问虚地址256H, D65H, 6B7H各需多少时间?

【解析】 (1)页面大小为1 KB, 占10位, 256H = 0010 0101 0110对应的页号 为0, 0号页在页表中, 不在快表中, 则需时间10ns+50ns+50ns=110(ns);

1. 【南京理工大学 2017】请求分页管理系统中,页面大小为1KB,采用固定分配局部置换策略。访存一次时间为50ns,访快表一次的时间为10ns,处理一次缺页的平均时间为50ms(已包括更新TLB和页表的时间)。设快表初始为空,快表未命中时,忽略快表的更新时间。系统为某进程分配3个页框,其页表如表所示。设该进程有逻辑地址访问序列256H,D65H,AB7H,456H,6B7H,若采用CLOCK置换算法(查询被置换页的指针初始在0号页,查找顺序按表由上至下)。

页号	页框号	访问位
0	6	1
1	7	0
2	4	0

(1)请问遵循上述地址访问序列,访问虚地址256H, D65H, 6B7H各需多少时间?

【解析】 (1)页面大小为1 KB, 占10位, D65H=1101 0110 0101对应的页号为3, 3号页不在页表中, 也不在快表中, 则需时间为10ns+50ns+50 ms+10ns+50ns=50000120(ns);

1.【南京理工大学 2017】请求分页管理系统中,页面大小为1KB,采用固定分配局部置换策略。访存一次时间为50ns,访快表一次的时间为10ns,处理一次缺页的平均时间为50ms(已包括更新TLB和页表的时间)。设快表初始为空,快表未命中时,忽略快表的更新时间。系统为某进程分配3个页框,其页表如表所示。设该进程有逻辑地址访问序列256H,D65H,AB7H,456H,6B7H,若采用CLOCK置换算法(查询被置换页的指针初始在0号页,查找顺序按表由上至下)。

页号	页框号	访问位
0	6	1
1	7	0
2	4	0

(1)请问遵循上述地址访问序列,访问虚地址256H, D65H, 6B7H各需多少时间?

【解析】 (1)页面大小为1 KB,占10位,6B7H=0110 1011 0111B对应的页号为1,1号页在页表中,不在快表中,则需时间为10ns+50ns+50ns=110(ns)。

1.【南京理工大学 2017】请求分页管理系统中,页面大小为1KB,采用固定分配局部置换策略。访存一次时间为50ns,访快表一次的时间为10ns,处理一次缺页的平均时间为50ms(已包括更新TLB和页表的时间)。设快表初始为空,快表未命中时,忽略快表的更新时间。系统为某进程分配3个页框,其页表如表所示。设该进程有逻辑地址访问序列256H,D65H,AB7H,456H,6B7H,若采用CLOCK置换算法(查询被置换页的指针初始在0号页,查找顺序按表由上至下)。

页号	页框号	访问位
0	6	1
1	7	0
2	4	0

(2)遵循题目中给出的地址访问序列,写出逻辑地址256H,D65H,456H对应的物理地址 (十六进制表示)。

【解析】逻辑地址256H = 0110 0101 0110的页号为0,对应的块号为6,则物理地址为 011010 0101 0110=1A56H;

1. 【南京理工大学 2017】请求分页管理系统中,页面大小为1KB,采用固定分配局部置换策略。访存一次时间为50ns,访快表一次的时间为10ns,处理一次缺页的平均时间为50ms(已包括更新TLB和页表的时间)。设快表初始为空,快表未命中时,忽略快表的更新时间。系统为某进程分配3个页框,其页表如表所示。设该进程有逻辑地址访问序列256H,D65H,AB7H,456H,6B7H,若采用CLOCK置换算法(查询被置换页的指针初始在0号页,查找顺序按表由上至下)。

页号	页框号	访问位
0	6	1
1	7	0
2	4	0

(2)遵循题目中给出的地址访问序列,写出逻辑地址256H,D65H,456H对应的物理地址 (十六进制表示)。

【解析】逻辑地址D65H=1101 0110 0101对应的页号为3,淘汰1号页,将3号页替换1号页,3号页对应的块号为7,则物理地址为011101 0110 0101=1D65H;

1. 【南京理工大学 2017】请求分页管理系统中,页面大小为1KB,采用固定分配局部置换策略。访存一次时间为50ns,访快表一次的时间为10ns,处理一次缺页的平均时间为50ms(已包括更新TLB和页表的时间)。设快表初始为空,快表未命中时,忽略快表的更新时间。系统为某进程分配3个页框,其页表如表所示。设该进程有逻辑地址访问序列256H,D65H,AB7H,456H,6B7H,若采用CLOCK置换算法(查询被置换页的指针初始在0号页,查找顺序按表由上至下)。

页号	页框号	访问位
0	6	1
1	7	0
2	4	0

(2)遵循题目中给出的地址访问序列,写出逻辑地址256H,D65H,456H对应的物理地址 (十六进制表示)。

【解析】逻辑地址456H = 0100 0101 0110对应的页号为1, 1号页不在内存, 2号页淘汰, 将1号页替换2号页,则1号页对应的块号为4,则物理地址为010000 0101 0110=1056H。

页面大小为4 KB,一次内存的访问时间是60ns,一次快表(TLB)的访问时间是10ns,处理一次缺页的平均时间为2000ns(已含更新TLB和页表的时间)。进程的驻留集大小固定为2,采用最近最久未使用页面置换算法(LRU)和局部淘汰策略,假设

①TLB初始为空

- ②地址转换时先访问TLB, 若TLB未命中, 再访问页表 (忽略访问页表之后的TLB更新时间)
- ③有效位为0表示页面不在内存,产生缺页中断,缺页中断后已获得新调入页所在的页框号, 无须再访问页表或快表,

设有虚地址访问序列23E2H, 1566H, 2CB7H。请问:

(1)依次访问上述三个虚地址, 各需多少时间? 给出计算过程。

【解析】(1)页面大小为4KB,占12位,23E2H = 0010 0011 1110 0010的页号为2,2号页不在快表中,在页表中,则需要时间为10ns+60ns+60ns=130(ns);

页号	页框号	访问位
0	6	1
1	7	0
2	4	0

页面大小为4 KB,一次内存的访问时间是60ns,一次快表(TLB)的访问时间是10ns,处理一次缺页的平均时间为2000ns(已含更新TLB和页表的时间)。进程的驻留集大小固定为2,采用最近最久未使用页面置换算法(LRU)和局部淘汰策略,假设

①TLB初始为空

- ②地址转换时先访问TLB,若TLB未命中,再访问页表(忽略访问页表之后的TLB更新时间)
- ③有效位为0表示页面不在内存,产生缺页中断,缺页中断后已获得新调入页所在的页框号, 无须再访问页表或快表,

设有虚地址访问序列23E2H, 1566H, 2CB7H。请问:

(1)依次访问上述三个虚地址, 各需多少时间? 给出计算过程。

【解析】(1)页面大小为4KB,占12位,1566H = 0001 0101 0110 0110的页号为1,1号页不在内存中,则需要时间为60ns+2000ns+10ns+60ns=2130(ns)

页号	页框号	访问位
0	6	1
1	7	0
2	4	0

页面大小为4 KB,一次内存的访问时间是60ns,一次快表(TLB)的访问时间是10ns,处理一次缺页的平均时间为2000ns(已含更新TLB和页表的时间)。进程的驻留集大小固定为2,采用最近最久未使用页面置换算法(LRU)和局部淘汰策略,假设

①T	LB初	1始)	与空
(L)	ロレル.	ノグロノ	ىلەن

- ②地址转换时先访问TLB, 若TLB未命中, 再访问页表 (忽略访问页表之后的TLB更新时间)
- ③有效位为0表示页面不在内存,产生缺页中断,缺页中断后已获得新调入页所在的页框号, 无须再访问页表或快表,

设有虚地址访问序列23E2H, 1566H, 2CB7H。请问:

(1)依次访问上述三个虚地址, 各需多少时间? 给出计算过程。

【解析】(1)页面大小为4KB,占12位,2CB7H = 0010 1100 1011 0111的页号也为2,2 号页在快表中,也在页表中,则时间为10ns+60ns=70(ns)。

页号	页框号	访问位
0	6	1
1	7	0
2	4	0

页面大小为4 KB,一次内存的访问时间是60ns,一次快表(TLB)的访问时间是10ns,处理一次缺页的平均时间为2000ns(已含更新TLB和页表的时间)。进程的驻留集大小固定为2,采用最近最久未使用页面置换算法(LRU)和局部淘汰策略,假设

页号	页框号	访问位
0	6	1
1	7	0
2	4	0

- ①TLB初始为空
- ②地址转换时先访问TLB,若TLB未命中,再访问页表(忽略访问页表之后的TLB更新时间)
- ③有效位为0表示页面不在内存,产生缺页中断,缺页中断后已获得新调入页所在的页框号, 无须再访问页表或快表,

页号	页框号	访问位
0	6	1
1	7	0
2	4	0

设有虚地址访问序列23E2H, 1566H, 2CB7H。请问:

(2)基于上述访问序列,虚地址1566H的物理地址是多少?请说明理由。

【解析】采用最近最久未使用页面置换算法, 0001 0101 0110 0110的页号为 1, 1号页不在内存中,则淘汰0号页,将1号页调入内存,此时1号页的地址为 2C8H,则虚地址1566H的物理地址是2C8566H。

页号	页框号	访问位
0	6	1
1	7	0
2	4	0

3. 页式存储管理允许用户的编程空间为32个页面 (每页1KB), 主存为16 KB。如有一用户程序有10页长, 且某时刻该用户程序页表如表所示。若分别遇到三个逻辑地址0AC5H, 1AC5H, 3AC5H的操作, 试计算并说明存储管理系统将如何处理。

逻辑页号	物理块号	
0	8	
1	7	
2	4	
3	10	

4、有一个虚拟存储系统,物理内存共有3页,其中一页用来存放程序,其余两页用于存放数据。假设程序已在内存中占一页,其余两页空闲,且矩阵是行优先存储。

程序A:

```
for(i=0;i<100;i++)

for(j=0; j<100; j++)

A[i,j] =0;

B:
```

程序B:

for(j=0; j<100; j++) for(i=0;i<100;i++) A[i, j] =0;

若每页可存放200个整数,程序A和程序B的执行过程各会发生多少次缺页?若每页只能存放100个整数,程序A和程序B的执行过程各会发生多少次缺页?以上说明了什么问题?

5. 某一个计算机系统采用虚拟页式存储管理方式,当前在处理器上执行的某一个进程的页表如表所示,所有的数字均为十进制,每一项的起始编号是0,并且所有的地址均按字节编址,每页的大小为1024 B。

逻辑页号	存在位	引用位	修改位	页框号
0	1	1	0	4
1	1	1	1	3
2	0	0	0	
3	1	0	0	1
4	0	0	0	
5	1	0	1	5

(1)将下列逻辑地址转换为物理地址,写出计算过程,对不能计算的说明其原因。

0793, 1197, 2099, 3320, 4188, 5332

5. 某一个计算机系统采用虚拟页式存储管理方式,当前在处理器上执行的某一个进程的页表如表所示,所有的数字均为十进制,每一项的起始编号是0,并且所有的地址均按字节编址,每页的大小为1024 B。

逻辑页号	存在位	引用位	修改位	页框号
0	1	1	0	4
1	1	1	1	3
2	0	0	0	
3	1	0	0	1
4	0	0	0	
5	1	0	1	5

(2)假设程序欲访问第2页,页面置换算法为改进的CLOCK算法,请问该淘汰哪页?页表该如何修改?上述地址的转换结果是否改变?变成多少?

- 6. 设一进程共有5页,其中程序占3页,常数占1页,工作单元占1页,它们依次存放在外存的第45,46,98,99和100块。现在程序段已分配在内存的第7,10,19页,而常数区和工作区尚未获得内存,请回答下述问题:
- (1)页表应包括哪些项目?填写此页表。

- 6. 设一进程共有5页,其中程序占3页,常数占1页,工作单元占1页,它们依次存放在外存的第45,46,98,99和100块。现在程序段已分配在内存的第7,10,19页,而常数区和工作区尚未获得内存,请回答下述问题:
- (2)在运行过程中因需要使用常数而发生中断,假定此时内存无空闲页面,需要把第9页淘汰,操作系统应如何处理?页表又将发生什么变换?

- 7. 【广东工业大学 2014】在分页存储管理系统中,存取一次内存的时间是8ns,查询一次快表的时间是1ns,缺页中断的时间是20ns。假设页表的查询与快表的查询同时进行,当查询页表时,如果该页在内存但快表中没有页表项,系统将自动把该页页表项送入快表。一个作业最多可保留3个页面在内存。现在开始执行一作业,系统连续对作业的2,4,5,2,7,6,4,8页面的数据进行一次存取,如分别采用FIFO算法和最优页面置换算法,对每种置换算法求:
- (1)存取这些数据需要的总时间。

- 7. 【广东工业大学 2014】在分页存储管理系统中,存取一次内存的时间是8ns,查询一次快表的时间是1ns,缺页中断的时间是20ns。假设页表的查询与快表的查询同时进行,当查询页表时,如果该页在内存但快表中没有页表项,系统将自动把该页页表项送入快表。一个作业最多可保留3个页面在内存。现在开始执行一作业,系统连续对作业的2,4,5,2,7,6,4,8页面的数据进行一次存取,如分别采用FIFO算法和最优页面置换算法,对每种置换算法求:
- (2)产生缺页中断的次数(包括初次调入)。

8. 【燕山大学 2012】在分页存储管理系统中,页面的大小为1KB,存取一次内存的时间是10µs,查询一次快表的时间是1µs,缺页中断的时间是1000µs。现开始执行一作业,连续访问0E3F,134F,1720,0E4A,1E41,1811,136A,0E10逻辑地址上的数据。假设:一个作业最多可保留3个页面在内存,页表的查询与快表的查询同时进行,快表容量足够大,当查询页表时,如果该页在内存但快表中没有页表项,系统将自动地把该页页表项送入快表。如分别采用FIFO和OPT(最优页面置换算法),试求:

(1)这个作业依次访问了哪些逻辑页面?

- 8.【燕山大学 2012】在分页存储管理系统中,页面的大小为1KB,存取一次内存的时间是10µs,查询一次快表的时间是1µs,缺页中断的时间是1000µs。现开始执行一作业,连续访问0E3F,134F,1720,0E4A,1E41,1811,136A,0E10逻辑地址上的数据。假设:一个作业最多可保留3个页面在内存,页表的查询与快表的查询同时进行,快表容量足够大,当查询页表时,如果该页在内存但快表中没有页表项,系统将自动地把该页页表项送入快表。如分别采用FIFO和OPT(最优页面置换算法),试求:
- (2)每种算法各发生几次缺页中断?

- 8. 【燕山大学 2012】在分页存储管理系统中,页面的大小为1KB,存取一次内存的时间是10µs,查询一次快表的时间是1µs,缺页中断的时间是1000µs。现开始执行一作业,连续访问0E3F,134F,1720,0E4A,1E41,1811,136A,0E10逻辑地址上的数据。假设:一个作业最多可保留3个页面在内存,页表的查询与快表的查询同时进行,快表容量足够大,当查询页表时,如果该页在内存但快表中没有页表项,系统将自动地把该页页表项送入快表。如分别采用FIFO和OPT(最优页面置换算法),试求:
- (3)每种算法下,每次数据访问需要的时间? (需列式子)

- 9. 【西南大学 2018,西北大学 2018】在一个采用页式虚拟存储管理的系统中,有一用户作业,它依次要访问的字地址序列是115,228,120,88,446,102,321,432,260,167,若该作业的第0页已经装入主存,现分配给该作业的主存共300字,页的大小为100字,请回答下列问题:
- (1)现分配给该作业的主存为3,按FIFO调度算法和LRU调度算法将分别产生多少次缺页中断,缺页中断率为多少?

- 9.【西南大学 2018,西北大学 2018】在一个采用页式虚拟存储管理的系统中,有一用户作业,它依次要访问的字地址序列是115,228,120,88,446,102,321,432,260,167,若该作业的第0页已经装入主存,现分配给该作业的主存共300字,页的大小为100字,请回答下列问题:
- (2)现分配给该作业的主存块为4,按FIFO调度算法和LRU调度算法将分别产生多少次缺页中断,缺页中断率为多少?

谢谢大家

考点五:

带快表的两级系统的地址翻译过程

【政哥点拨】

1. 某计算机主存按字节编址,采用二级分页存储管理,地址结构如下:页目录号(10位)页号(10位)页内偏移(12位)

虚拟地址2050 1225H对应的页目录号、页号分别是()。

A. 081H,101H B. 081H,401H C. 201H,101H D. 201H,401H

A题中给出的是十六进制地址,首先将它转换为二进制地址,然后用二进制地址去匹配题中对应的地址结构。转换为二进制地址和地址结构的对应关系如下所示。

【牛刀小试】



