

Part 2 连续分配方式

● 问题

1. 固定分区有何缺点?
2. 可变分区如何分配内存?
3. 对换和覆盖技术有何作用?
4. 可变分区分配有哪些算法?
5. 分页如何分配内存?



学习目标

- 能够理解分页存储管理的原理
- 掌握并应用分页存储的地址变换过程
- 理解快表的作用以及有快表的地址变换过程
- 理解多级页表、反置页表
- 能够分析分页存储管理的特点

Part 3 分页存储管理

3.1 分页存储基本原理

Frame
Number

0	
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	

0	A.0
1	A.1
2	A.2
3	A.3
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	

0	A.0
1	A.1
2	A.2
3	A.3
4	B.0
5	B.1
6	B.2
7	
8	
9	
10	
11	
12	
13	
14	

Part 3 分页存储管理

3.1 分页存储基本原理

0	A.0
1	A.1
2	A.2
3	A.3
4	B.0
5	B.1
6	B.2
7	C.0
8	C.1
9	C.2
10	C.3
11	
12	
13	
14	

0	A.0
1	A.1
2	A.2
3	A.3
4	
5	
6	
7	C.0
8	C.1
9	C.2
10	C.3
11	
12	
13	
14	

0	A.0
1	A.1
2	A.2
3	A.3
4	D.0
5	D.1
6	D.2
7	C.0
8	C.1
9	C.2
10	C.3
11	D.3
12	D.4
13	
14	

Part 3 分页存储管理

3.1 分页存储基本原理

内存地址 (逻辑地址) 范围 (1块的大小为1KB) :

0#块: 0 ~ 1023 (1K-1)

0000 0000 0000 ~ 0011 1111 1111

.....

1#块: 1024~2047 (2K-1)

0100 0000 0000 ~ 0111 1111 1111

2#块: 2048 ~3071 (3K-1)

1000 0000 0000 ~ 1011 1111 1111

3#块: 3072 ~4095 (4K-1)

1100 0000 0000 ~ 1111 1111 1111

n#块: $n \times 1k \sim (n+1) \times 1K - 1$

**00 0000 0000 ~ **11 1111 1111

内存地址地址结构:

** 00 0000 0000

Part 3 分页存储管理

3.2 地址结构

- 地址结构与数对（页号，页内位移）的形成

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	1	0	1	1	1	0	1	1	1	0	0	0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	1	0	1	1	1	0	1	1	1	0	0	0
← 页号 →						← 页内地址 →									

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	1	0	1	1	1	0	1	1	1	0	0	0
← 页号 →								← 页内地址 →							

Part 3 分页存储管理

3.2 地址结构

31 12 11 0

页号P

位移量W (页内地址)

分页地址中的结构 (32位)

页号P

- ◆ 12-31位: 20位
- ◆ 地址空间最多允许有1M (2^{20}) 页

位移量W (页内地址)

- ◆ 0-11: 12位
- ◆ 每页大小为4KB (2^{12})

Part 3 分页存储管理

3.2 地址结构

对某特定机器，地址结构是一定的。

若给定一个逻辑地址空间中的地址为A，页面的大小为L，则页号P和页内地址d可按式求得

$$P = \text{INT} \left[\frac{A}{L} \right]$$

$$d = [A] \text{ MOD } L$$

分页方式要解决的问题：

- 1.程序在内存的位置如何记录？
- 2.地址转换如何完成？
- 3.时间效率分析

Part 3 分页存储管理

3.3 数据结构

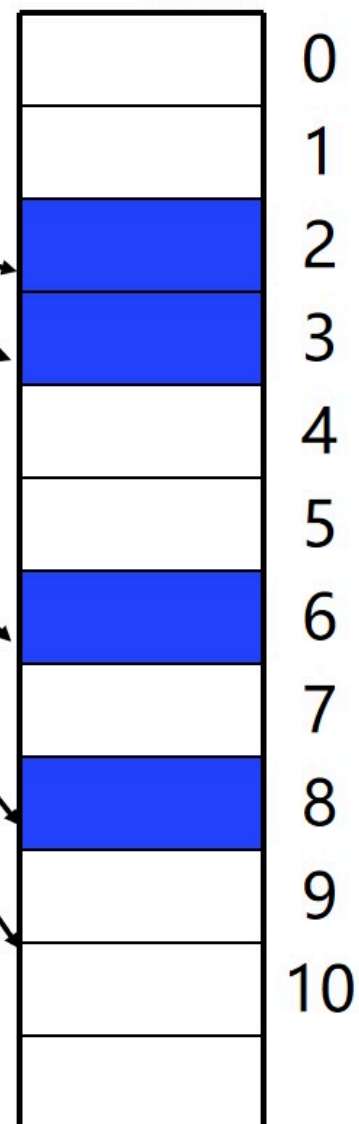
用户程序

0页
1页
2页
3页
4页
5页
.....
n页

页表

页号	块号
0页	2
1页	3
2页	6
3页	8
4页	9
5页	
...	...
n页	

内存



Part 3 分页存储管理

3.3 数据结构

- **进程页表**：每个进程有一个页表，描述该进程占用的物理页面及逻辑排列顺序；

 - ❖ 逻辑页号（本进程的地址空间） - > 物理页面号（实际内存空间）；

- **物理页面表**：整个系统有一个物理页面表，描述物理内存空间的分配使用状况。

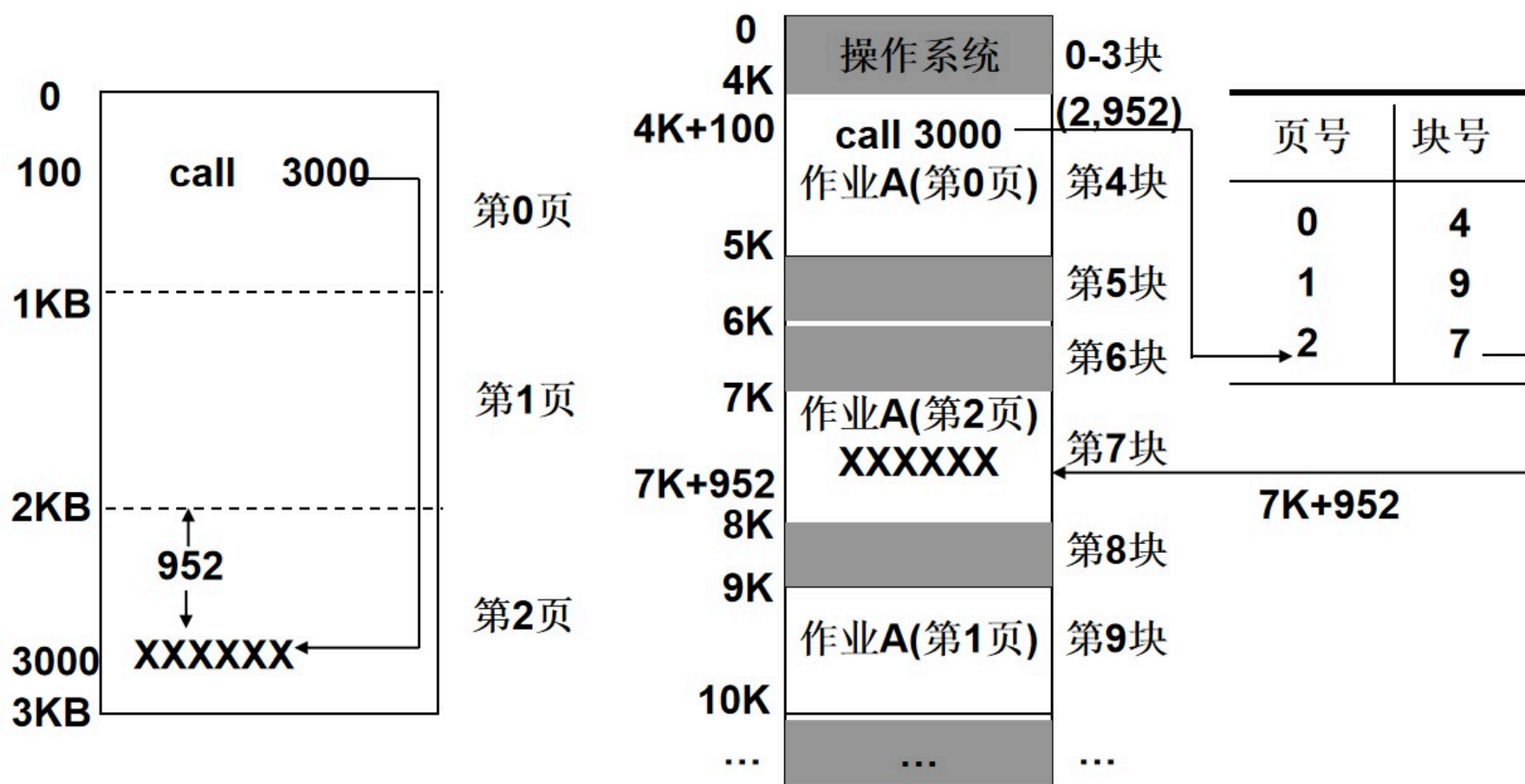
 - ❖ 数据结构：位示图，空闲页面链表；

- **请求表**：整个系统有一个请求表，描述系统内各个进程页表的位置和大小，用于地址转换，也可以结合到各进程的PCB里。

Part 3 分页存储管理

3.4 地址变换

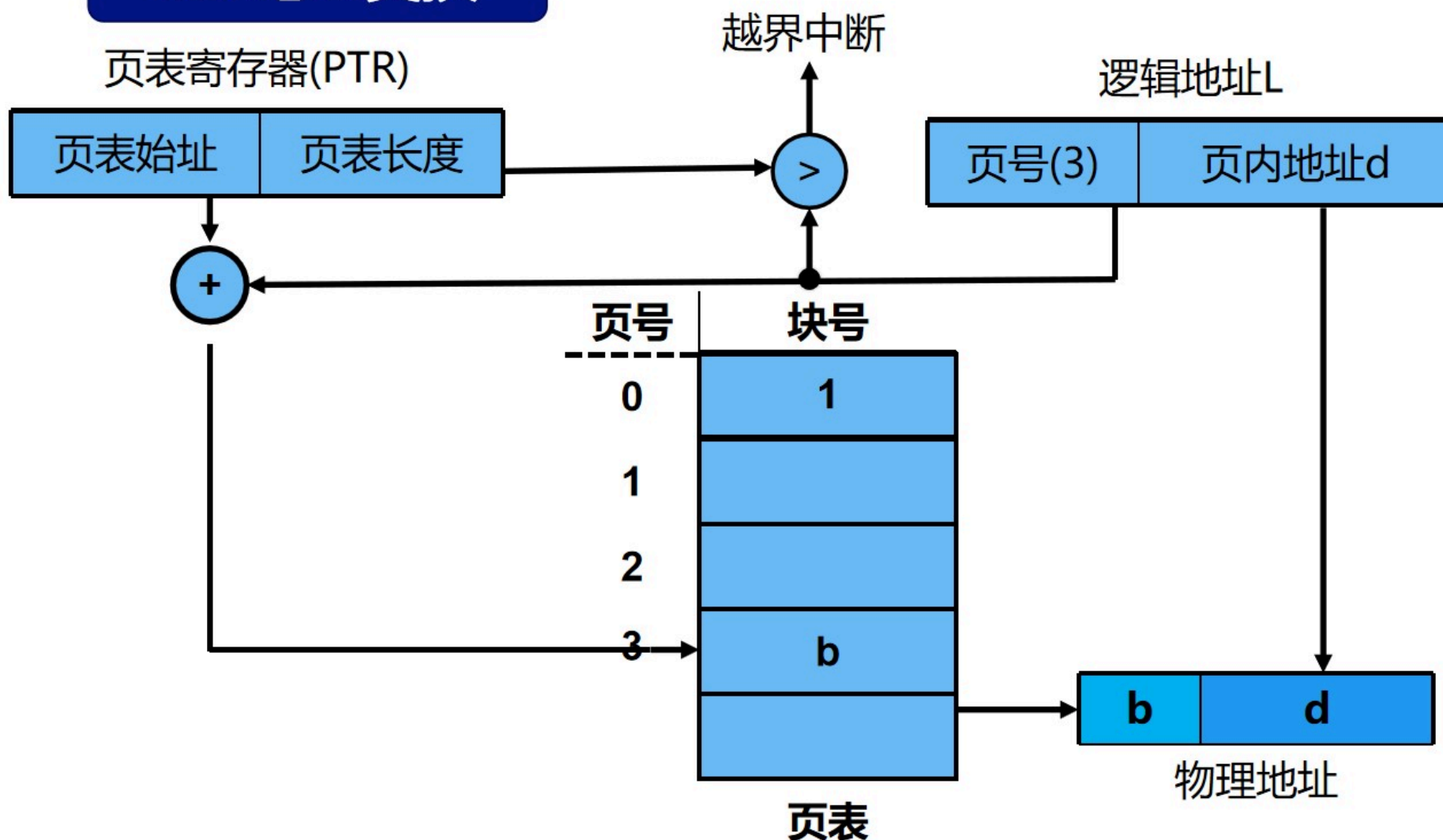
分页存储管理中程序的执行



Part 3 分页存储管理

3.4 地址变换

基本的地址变换机构



3.4 地址变换

地址变换例题1

- 某分页存储管理系统的每页大小2KB，内存为256KB。假定某时刻一用户页表如下：
- 逻辑地址053CH（十六进制）和142FH（十六进制）所对应的物理地址是什么？

页号	物理块号
0	5
1	0
2	9
3	7

3.4 地址变换

地址变换例题2

- 某分页存储管理系统，页大小为4KB，用户程序装入程序后的页表如下，分析程序指令：Load A，5200如何执行，该指令执行时访问几次内存？

页号	块号
0	5
1	9
2	4
3	12

3.4 地址变换

特点

- 增加了指令执行时间：页表在内存，一条指令执行至少需要访问两次内存。
- 缺点：指令执行时间延长

3.5 相关技术

1.快表

●1.快表

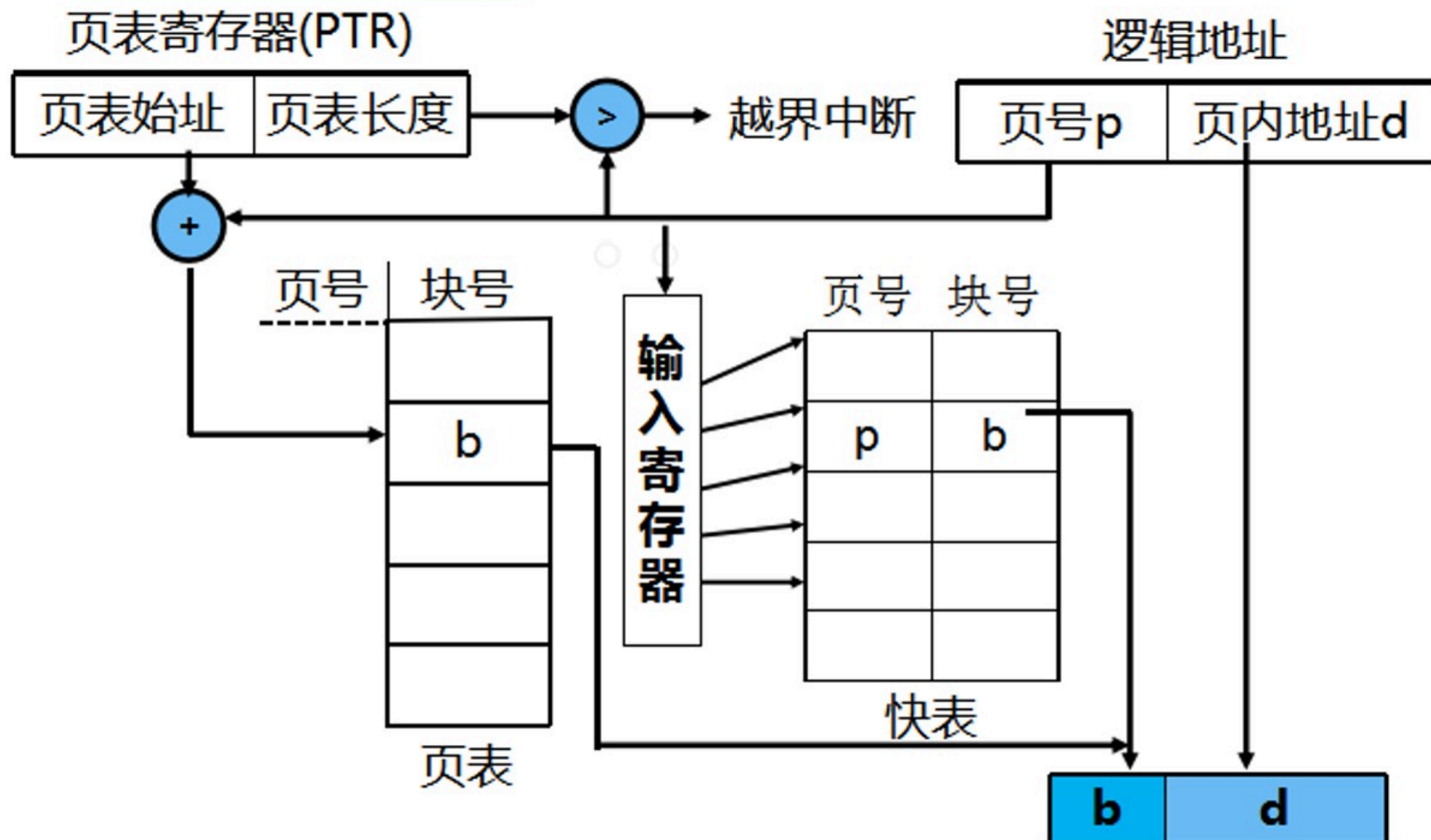
❖为缩短查找时间，可以将页表从内存装入到相联存储器(TLB, Translation Lookaside Buffer)，按内容查找(associative mapping)，即逻辑页号
- > 物理页号

页号	块号	访问位	状态位

Part 3 分页存储管理

3.5 相关技术

1. 快表 具有快表的地址变换机构



3.5 相关技术

2. 内存有效访问时间EAT

- ❖ 访问一次内存的时间为 t ，快表命中率为 a ，快表访问一次时间为 λ
- ❖ 基本分页存储管理 $EAT = t + t = 2t$
- ❖ 有快表的分页存储管理方式
- ❖ $EAT = a \times \lambda + (t + \lambda) \times (1 - a) + t = 2t + \lambda - t \times a$

3.5 相关技术

2.内存有效访问时间EAT举例

- 假定访问主存时间为100毫微秒，访问相联存储器时间为20毫微秒，相联存储器为32个单元时快表命中率可达90%，按逻辑地址存取的平均时间为：
$$(100 + 20) \times 90\% + (100 + 100 + 20) \times (1 - 90\%) = 130 \text{ 毫微秒}$$
- 两次访问主存的时间：
$$100 \text{ 毫微秒} \times 2 = 200 \text{ 毫微秒。}$$