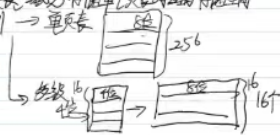


5-16 5. 1)  $2^{12}$  KB 空间  $(2^{12}/8) \times 8 = 2^{12}$   $(2^{12}/16) \times 16 = 2^{12}$   
 2)  $2^{26}$  KB 空间  $48+3-25 = 51-25 = 26$

3) 因为多次页表相对页表, 减少了有能单页表对应的存储空间  
 例如 8位索引  $\rightarrow$  页表



举个例子: 1个  $2^{26}$  页的程序  
 页表:  $8 \times 2^8 = 2^{11}$

多级表: 1个2级表:  $4 \times 2^8 + 4 \times 2^8 = 2^9$   
 (一个表有内容才占用)

6. 中间位的数字变化更快, 能够在组中作更多的变化, 能对组的空间更加有效利用, 一个组作为整体调用, 因此, 变化的高位在标号更组  
 如果高位作组索引, 对组的空间不能有效利用, 而组则需要高频切换, 不适合  
 因此, 中间位作组索引, 高位作标志。

7. 对页表缓存中的一个标志对应的, 列表表与虚拟存储的页相对应, 缓存就能在一页中切换, 而对应的物理地址仍在同一组, 有连续性, 不用再读虚拟表来切换不同页表。

8. (1)  $1 \times 77\% + 10 \times 3\% = 0.97 + 3\% = 4.27$  页

(2)  $L_0 = 2^6 \text{ MB} = 2^{20} \text{ KB}$   
 $L_1 = 64 \text{ KB} = 2^6 \text{ KB}$   
 $\therefore$  随机  
 $\therefore$  命中率  $= \frac{2^6}{2^{20}} = \frac{1}{2^{14}}$   
 $1 \times \frac{1}{2^{14}} + 10 \times \frac{1}{2^{14}} = 10$

(3) 只有当缓存存在程序的所访问的主存中大部分时, 用程序在局部空间中连续, 而此时缓存才能起较好的作用。

(4) 设  $x$   
 $1 \cdot x + 10 \cdot (1-x) < 105$   
 $-10x < -5$   
 $x > \frac{1}{2}$   
 当  $L_1$  的命中率大于  $\frac{1}{2}$  时, 应用  $L_1$  才是有意义的

9.

编号	地址位数 Bit	缓存大小 KB $2^{10}$	块大小 Byte	相联度	组数量	组索引位数 Bit	标签位数 Bit	偏移位数 Bit
1	32	4 $2^2$	64 $2^6$	2 $2^1$	32	21	5	6
2	32	4	64	8 $2^3$	8	23	3	6
3	32	4	64	全相联	1	26	0	6
4	32	16 $2^4$	64 $2^6$	1	256	18	8	6
5	32	16	128 $2^7$	2	64	19	6	7
6	32	64 $2^{10}$	64	4 $2^2$	256	18	8	6
7	32	64	64	16 $2^4$	64	29	6	6
8	32	64	128 $2^7$	16 $2^4$	32	20	5	7

10. (1)  $0.22 \times (1-p_1) + 0.22 \times p_1 < 0.52 \times (1-p_2) + 0.52 \times p_2$   
 $100p_1 \cdot 0.22 < 100p_2 \cdot 0.52$   
 $p_1 - p_2 < \frac{3 \times 10^{-3}}{2}$   
 满足:  $p_1 < p_2 + 3 \times 10^{-3}$  时, 平均访问

(2)  $0.22 + k \cdot 0.22 p_1 < 0.52 + k \cdot 0.52 p_2$   
 $k(0.22p_1 - 0.52p_2) < 0.3$   
 $0.22p_1 - 0.52p_2 < \frac{0.3}{k}$   
 满足  $p_1 < \frac{0.3}{22k} + \frac{52}{22} p_2$  时, 平均访问

11.  $0x1001$   $0x1005$   $0x1021$   $0x1045$   $0x1305$   $0x2005$   $0x200b$   
 16块 直接映射: 4位 6位  
 32块 6位  
 256块 256映射: 8位 6位

4块映射: 2位 7位

8块映射: 1位 7位

12.  $256 \div 16 = 16$   $1 \text{ Byte} = 4 \times 32 \text{ bit}$

A: 8+8 B: 16

$\frac{3}{11} - \frac{1}{11}$

A: 每 8 次 miss - 1 次 96/32=3  
 B: 每 16 次 miss - 1 次 96/64=1.5  
 $\frac{3}{1.5} = \frac{1}{0.5}$   
 $\frac{96}{96} = \frac{1}{1}$

B < A

13) for (int j=0; j<(28; ++j))  
 for (int i=0; i<(64; ++i))  
 {  
 A[i][j] = A[j][i+1];  
 }

14) 1) 优化前  $2^2 \times 2^3 \times 2^3 = 2^8 = 256$  次访问  
 $2^2 \times 2^3 \times 2^3 = 2^8 = 256$  次访问  
 优化后  
 $2^2 \times 2^3 \times 2^3 = 2^8 = 256$  次访问  
 $2^2 \times 2^3 \times 2^3 = 2^8 = 256$  次访问

2) 优化前  $128 \times 8 = 2^{10}$   
 优化后  $8 \times 128 = 2^{10}$

3) 优化前  $8 \times 128 \times 22 \times 2^3 \times 2^3 \times 2^3 = 2^{10} \times 2^5 = 2^{15}$   
 $2^3 \times 2^3 = 2^6$   
 $2^3 \times 2^3 = 2^6$   
 $2^3 \times 2^3 = 2^6$

15)

```
int input[4][4], output[4][4];
for(int i=0; i<4; ++i){
  for(int j=0; j<4; ++j){
    output[j][i] = input[i][j];
  }
}
```

假设系统中 int 变量为 4 字节，input 数组的起始地址为 0x0，output 数组的起始地址为 0x40。假设上述程序运行在一个存在 L1 缓存的系统中，该缓存总大小 32 字节，块大小 16 字节，直接映射，缓存策略为直写、写分配。若初始缓存为空，系统仅在对 input 和 output 的读写时会访问缓存，请在下表中填写执行上述程序时数组中每个元素的缓存命中情况。



一块为 4 行 int  
 缓存共 8 行 int

	input 数组				output 数组			
	列 0	列 1	列 2	列 3	列 0	列 1	列 2	列 3
行 0	miss	miss	miss	miss	miss	miss	miss	miss
行 1	miss	miss	miss	miss	miss	miss	miss	miss
行 2	miss	miss	miss	miss	miss	miss	miss	miss
行 3	miss	miss	miss	miss	miss	miss	miss	miss

16) 1) 512B 块  $512/4=128$  块 2 路  
 $64+64$

共读 256 次数字  
 共 miss 4 次

命中率  $\frac{252}{256} = 0.984375$

2) 可以当缓存为 1KB 时  
 只 miss 2 次

3) 不能，因为不论块大小如何，在缓存大小和关联度不变时，能存入缓存的数据值总量不变。

