

No.

Date. 作业12

7. 可以方便地实现虚拟内存到物理内存的映射，可以直接使用地址的组索引和块内偏移来计算物理地址，提高访问速度。

6. 使用高位作为组索引，中间位作为标签的话，每组的大小会变得非常小，导致组相对较多，而每组的块数相对较少，这会导致缓存的利用率降低，同时影响缓存的命中率。反之，则可以更好利用缓存存储空间，提高缓存命中率。

$$8.(1) 3\% \times 110 + 97\% = 4.2\%$$

(2) 1GB 被划分为 64KB 的块

$$2^{20} / 2^6 = 2^{14} = 16384 \text{ 块}$$

命中率 $1 / 16384$ 缺失率: $1 - \frac{1}{16384}$

$$4.2\% \times \left(1 - \frac{1}{16384}\right) \approx 4.2\%$$

(3) 局部性原理：

空间局部性：程序在某一段时间内访问的数据通常在空间上是相邻的

时间局部性：程序在某一时间段内的数据通常是相同或相似的

数组在空间上相邻，通过缓存可以大大提高访存性能，使得访问数组的随机位置的平均访问延时几乎只取决于缓存。

(4) 平均缓存命中率是可大 $x + (1-x) \times 110 \leq 105$ $x > 4.6\%$

9. 组数量 组索引位数 标签位数 偏移位数

1	32	5	22	6
2	8	3	23	6
3	16	0	26	6
4	28	8	18	6
5	26	6	19	7
6	28	8	18	6
7	26	6	20	6
8	25	5	20	7



扫描全能王 创建

$$10. \Delta T_1 = 0.22 \times (1 - P_1) + 100 \times P_1$$

$$\Delta T_2 = 0.52 \times (1 - P_2) + 100 \times P_2$$

$$\Delta T_1 < \Delta T_2$$

$$\therefore 99.78P_1 - 99.48P_2 < 0.3$$

$$(2) 0.22(1 - P_1) + 0.2^2 k P_1 < 0.52(1 - P_2) + 0.5^2 k P_2$$

$$0.22 + 0.22(k-1)P_1 < 0.52 + 0.52(k-1)P_2$$

$$0.22P_1 - 0.52P_2 < \frac{0.3}{k-1}$$



扫描全能王 创建

块地址

	索引	块内偏移	直接映射：块内偏移6位 索引4位
0x100	0001 0000 0000 000	0x100	与 0x1005, 0x1021 2次替换
0x1005	0001 0000 000D 0101	0x1305	与 0xff05 1次替换
0x1021	0001 0000 0010 000	共3次替换	
0x1045	0001 0000 0100 0101		
0x1305	0001 0011 0000 0101	2路: $16/2 = 8$	3位索引
0x2ee5	0010 1110 1110 0101	000 000 000 00	100 011 100
0xff05	1111 1111 0000 0101	1次替换	

4路: $16/4 = 4$ 2位索引 00 00 00 01 00 11 00

1次替换

8路: 无替换



扫描全能王 创建

12. $256B - 2^8B - 0000000000$

16字节可存放 4个数组元素

直接映射 $a[0] a[1] a[2] a[3]$ $a[0]$ 缺失 $a[1] \sim a[3]$ 命中

缺失率 $\frac{1}{4}$

2路组相联 缺失率 $\frac{1}{8}$



扫描全能王 创建

```
13. for (int i=0; i<128; ++i) {  
    for (int j=0; j<64; ++j) }  
    A[i][j] = A[i][j] + 64;  
}  
}  
for (int i=0; i<64; ++i) {  
    for (int j=0; j<128; ++j) }  
    B[i][j] = A[j][i];  
}  
}  
for (int i=0; i<128; ++i) {  
    for (int j=0; j<64; ++j) }  
    B[i][j] = B[i][j] + 1;  
}  
}  
for (int i=0; i<64; ++i) {  
    for (int j=0; j<128; ++j) }  
    A[j][i] = B[i][j];  
}  
}
```



14. $4KB = 2^{12}$ 0000 0000 0000

① 每块 2^3 字节 可存放 8 个数(即元素)

优化前: $a[0][0], a[0][1], a[0][2], a[0][3], \dots, a[0][7]$
每次均缺失 $\star 64 \times 128 = 2^{14}$ 条 \Rightarrow 且每 T 数据访问 2^3

优化后: $a[0][0], \dots, a[0][7]$
访问 14 次中, 仅写入 $a[0][0]$ 时缺失 $\frac{2^{14}}{14}$ 次缺失

② 优化前 $\star 64 \times 128 \times 2 = 2^{14}$ 次

优化: 0 次

③ $\star 64 \times 128 = 2^{13}$ 容量



Date.

15. 每 2 块 额块 16 字节 可存 4 个数

#

input	output
0 1 2 3	0 1 2 3
0 miss miss miss miss	miss miss miss miss
1 hit hit hit hit	miss miss miss miss
2 hit hit hit hit	miss miss miss miss
3 hit hit hit hit	miss miss miss miss



扫描全能王 创建

No. _____

Date.

16.11) $512 / 16 = 32$, $32 / 2 = 16$ 块 $16 / 4 = 4$ 个数组元素
命中率 $\frac{1}{4}$

(2) 是的。缓存大小增大可以存储更多的数据，从而提高命中率

(3) 不一定。块大小增大可提高命中率，但也可能导致更多的缓存未命中

更多的数据。



扫描全能王 创建