

6. 解：为了避免冲突，如果使用高位作为组索引，由于高位的取值范围较小，容易导致多个不同的地址被映射到同一个组中，造成冲突。

7. 解：可以使缓存系统的管理与虚拟内存管理保持一致，从而提高系统的整体性能和稳定性。

8. 1)解：缺失惩罚 = 110 - 1 = 109 (周期)

$$\text{平均访问延时} = 1 + 109 \times 3\% = 4.27 \text{ (周期)}$$

2)解：1GB = 2³⁰KB

$$\text{缓存缺失率} = 1 - \frac{64}{2^{20}} = 1 - 2^{-14}$$

$$\begin{aligned}\text{平均访问延时} &= 1 + 109 \times (1 - 2^{-14}) \\ &\approx 110 \text{ (周期)}\end{aligned}$$

3)解：局部性原理可以使缓存命中率提高，减少缓存缺失带来的影响，从而使程序运行效率更高。

(4)解：令平均缓存命中率为 η 。

即有

$$1 + (1 - \eta) \times 109 < 105$$

$$\eta > \frac{5}{109} \approx 4.59\%$$

9	组数量	组差引数 Bit	校签位数 Bit	偏移位数 Bit
1.	32	5	21	6
2.	8	3	23	6
3.	1	0	26	6
4.	256	8	18	6
5.	64	6	19	7
6.	256	18	18	6
7.	64	6	20	6
8	32	5	20	7

10. 1) 解: $T_1 = 0.22(1-P_1) + 109P_1 = 0.22 + 99.78P_1$
 2) 解. $0.22 + 0.22K \cdot P_1 < 0.52 + 0.52K \cdot P_2$.
 即 $P_1 < \frac{15}{11(K-1)} + \frac{26}{11}P_2$

11. 角解. 直接映射

0x1001 0x1021 替换 1 次

0x1005 0x1045. 0x1305. 0x2005, 0xf+05.
 替换 4 次. 共 5 次.

乙路细颗粒 前 2 次不替换

后 3 次替换 3 次 共 3 次

4...: 4 组 替换了 3 次.

8...: 共 2 组, 不替换.

12. 缓存 A:

array[0]~[3], [4]~[7] ... [92]~[95]

每个数组元素存入一个缓存块中

且 $\text{array}[0] \sim [3] \dots [64] \sim [67] \rightarrow$ 第 1 块缓存

$\text{array}[28] \sim [31] \dots [92] \sim [95] \rightarrow$ 第 2 块缓存

放入 $[64] \sim [95]$ 及 $[20] \sim [31]$ 缓存

$[32] \sim [63]$ 不变

缺失率 = 2/3.

B. 3 行 8 组

每个都是刷新后访问

缺失率 = 1

13. 解: $\text{for } (\text{int } j=0; j < 128; ++j) \{$
 $\quad \text{for } (\text{int } i=0, i < 64, ++i)$
 $\quad \quad A[j][i] = A[j][i] + 1$

}

14. 1) 解: $4KB = 4 \times 2^{10} B = 2^{12} B$.

且 $\frac{2^{12}}{32} = 2^7 B$ 存储有 815 拼签

$128 \times 64 \approx 2^{13}$ 个数据 64 或 2^6 个数据

假设 A 篓, $2^7 / 2^6 = 2$. 每 2 个数据发生缺

失. $2^{13} / 2 = 2^{12}$ 次

LEIA 后: 1) 读取不缺失: 缓存坏块时读其

127 次

(2) 代價高。127R

代價低。63R.

3) 代價高。 $4 \times 64 = 256KB$

代價低， $4KB$

	input				output			
	0	1	2	3	0	1	2	3
0	miss	p	p	p	miss	miss	miss	miss
1	miss	p	p	p	p	p	p	p
2	miss	p	p	p	p	p	p	p
3	miss	p	p	p	p	p	p	p

$$\text{命中率} = \frac{15}{20} = 75\%.$$

(2) 因為緩存越大+越大後，提高命中率的
內存中的數據變多，命中率也之增加。
命中率增大。

(3) 33比44快的原因一個內存數據。