

$$5. 1) 2^{12} \text{ KB 空间} \quad (2^6 / 8) / 4 \times 8 / 4 = 2^{12} / 2^3 / 2^3 / 2^3 = 2^{12}$$

$$2) 2^{16} \text{ KB 空间} \quad 48 + 3 - 25 = 51 - 25 = 26$$

3) 因多次页表中页表减少了有效页数对应的物理空间
例如 8位索引 → 页表



举个例子：1个 2^6 页的程序
页表：8 * 2^8 = 2^11

$$\text{页表：} 1 \text{ 页表： } 4 \times 2^4 + 4 \times \frac{2^4}{2^3} \times 8 = 2^9$$

(一个表有相当大空间)

6. 中间位的教导变化更快，能够在操作中更快地变化，能对内存空间更有效地利用。一个组作为基本词元，因此变化的高位并不适合。
如果高位组变化，对进的空间不能有效利用，而组则需要
高位切换，不适合。
因此，中间位作组变化，高位作标志。

7. 对应后缓存中的一个地址对应的页表就与虚拟物理的页相对应，缓存就能在一页中切换，而对应的物理地址仍在其中，有连贯性，不用再读虚页表来切换不同的页。

$$8. (1) 1 \times 97\% + 110 \times 3\% = 0.97 + 3.3 = 4.27 \text{ 周期}$$

$$(2) L_1 = 2^6 \text{ MB} = 2^{20} \text{ KB}$$

L1：

$$\begin{aligned} & \because \text{命中率为 } \frac{25}{27} = \frac{1}{3} \\ & 1 \times \frac{1}{3} + 110 \times \frac{2}{27} = 110 \end{aligned}$$

(3) 只有当缓存在程序所访问的主存中占有大部分时，即程序在局部空间中连续，而此时缓存才能起较好作用。

(4) 错 X

$$1 \cdot x + 110 \cdot (1-x) < 105$$

$$-109x < -5$$

$x > \frac{109}{109}$
当 L1 的命中率大于 $\frac{109}{109}$ 时，应用 L1 才是有意义的

编号	地址位数 Bit	缓存大小 KB 2^3	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	64	2 ⁷	32	21	5	6
2	32	4	64	8 ²	8	23	3	6
3	32	4	64	全相联	1	26	0	6
4	32	$16 \cdot 2^4$	64	1	256	18	8	6
5	32	16	128	2	64	19	6	7
6	32	$64 \cdot 2^4$	64	$4 \cdot 2^8$	256	18	8	6
7	32	64	64	$16 \cdot 2^4$	64	20	6	6
8	32	64	128	$16 \cdot 2^4$	32	20	5	7

$$b. (1) 0.22 \times (1-p_1) + 10.22 \times p_1 < 0.52 \times (1-p_2) + 10.52 \times p_2$$

$$10.22 p_1 + 0.22 < 10.52 p_2 + 0.52$$

$$p_1 - p_2 < 3 \times 10^{-3}$$

满足 $p_1 < p_2 + 3 \times 10^{-3}$ 时，A 优于 B

$$(2) 0.22 + k \cdot 0.22 p_1 < 0.52 + k \cdot 0.52 p_2$$

$$k(0.22p_1 - 0.52p_2) < 0.3$$

$$0.22p_1 - 0.52p_2 < \frac{0.3}{k}$$

满足 $p_1 < \frac{0.3}{0.22} + \frac{0.52}{0.22} p_2$ 时，B 优于 A

$$11. 0x1001 \quad 0x1005 \quad 0x1021 \quad 0x1045 \quad 0x1055 \quad 0x2005 \quad 0x20f0$$

直接映射：4 组 64 次
3 组 64 次

2 组 64 次 2 路映射：3 组 64 次

4 路映射：2 组 7 次

8 路映射：1 组 7 次

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

$$A: 8+8 \quad B: 16$$

$$n = 8+8 = 16 \quad m = 16 \rightarrow \frac{3}{16} = \frac{1}{16}$$

$$A: \text{每8次 miss -> } 96 \div 32 = 3 \quad \frac{3}{96} = \frac{1}{32}$$

$$B: \text{每16次 miss -> } 96 \div 64 = 1 \dots \quad \frac{2}{96} = \frac{1}{48}$$

B/A

13. $\text{for}(\text{int } j=0; j < 28; ++j) \{$
 $\quad \text{for}(\text{int } i=0; i < 64; ++i) \{$
 $\quad \quad \}$

$$14. 1) \begin{array}{l} 2^{12} \times 2^5 \\ \text{优化前} \end{array} \quad 2^7 = 128 \quad \text{一块8个int}$$

$$128 \div 8 = 2^2 = 4 \rightarrow 16$$

$$2^4 \times 2^6 = 2^9 \text{块缺} \quad \boxed{2^8} \quad \boxed{15}$$

优化后
 2^{14} 但 $2^6 \times 2^3$ 次仅一次
 $\therefore 2^7$ 次缺失

$$2) \begin{array}{l} \text{优化前} \\ 128 \times 8 = 2^{10} \end{array} \quad \boxed{128} \quad \boxed{127 \dots 07}$$

优化后
 $8 \times 28 = 2^{10}$

$$3) \begin{array}{l} \text{优化前} \\ 8 \times 128 \times 32 = 2^3 \times 2^5 \times 2^5 = 2^{18} \times 2^5 \\ 32 \text{ KB} \end{array}$$

$$\begin{array}{l} \text{优化后} \\ 2^8 \times 2^3 = 2^8 \\ 256 \text{ B} \end{array}$$

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，缓存策略为写回、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 个 int
 共有 8 个 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

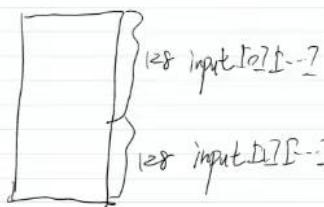
$$16 \times 4 \times 4 \text{ 块} \quad 512 \div 4 = 128 \text{ 块} \quad 2 \text{ 路}$$

共读 256 次数字
 共 miss 4 次

$$\therefore \text{命中率为 } \frac{252}{256} = 0.984375$$

b) 可以 当缓存为 1KB 时
 只 miss 2 次

(3) 不能，因为不论块大小如何，在缓存大小和无抖动不同时，
 有入缓存的数据总量不变。



128 input[0][0] - 7

128 input[1][1] - 7