5/17

5.(1) 页表大小4KB → 页内索引12位.

∴ 虚拟页号占 64-12 = 52位.

至多 2^{52} 页表项，1个页表项8位.

共 $2^{52} \times 2^{3} = 2^{35}$

即 33554432G

(2) $48-12=36$

$2^{36} \cdot 2^{3} = 2^{39} = 512G$

(3) 多级页表中一进程内存地址相对集中于连续.
随进程占用内存增大，才会增多该进程页表数目.
故页表占用空间小.

6. 一组数据连续存放时，相同高位下（含不同）中间位较多，使用高位索引易带来较多的缓存冲突. 而中位索引则不会那么多

7. 便于TLB与缓存并行访问. 选择虚拟地址中VPO低阶位可直接用于查找缓存.

8. (1) $3\% \times 110 + 97\% \times 1 = 4.27$

(2) $\frac{64KB}{1GB} = 0.0000610 4 \to 0.$

∴ 平均 $t_d \approx 110$

(3) 空间局部性使得数据块越小其利用率越低.
而时间局部性使得块越多其利用率越低
上述(2)的情况为空间局部性影响，块越小使命中率偏低

(4) $p + (1-p) \cdot 110 < 105.$

$p + 110 - 110p < 105$

$109p > 5$

$p > 4.59\%$

9.

编号	地址位数	缓存大小 KB	块大小 B	相联度	组数	索引位数	标记位数	偏移位数
1	32	4	64	2	32	5	21	6
2		4	64	8	8	3	23	6
3		4	64	全	1	0	26	6
4		16	64	1	256	8	18	6
5		16	128	2	64	6	19	7
6		64	64	4	256	8	18	6
7		64	64	16	64	6	20	6
8		64	128	16	32	5	20	7

10. (1) $P_1 \times 100.22ns + (1-P_1) \cdot 0.22ns < P_2 \cdot 100.52ns + (1-P_2) 0.52ns$

$$100.22P_1 + 22 - 22P_1 < 100.52P_2 + 52 - 52P_2$$

$$100.00P_1 < 100.00P_2 + 30.$$

$$P_1 < P_2 + 0.003$$

(2) $0.22ns + P_1 \cdot k \cdot 0.22ns < 0.52ns + P_2 \cdot k \cdot 0.52ns.$

$$22 + 22kP_1 < 52 + 52kP_2$$

$$22kP_1 < 52kP_2 + 30$$

$$11kP_1 < 26kP_2 + 15.$$

11. 16个块，块大小64 B，块内偏移6位。

地址(2进制为)

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1
0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1
0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

直接 16行块 16行组，4位索引，一组一个 5次替换

2路 16个块，8组，3位索引，一组2个，3次

4路 16块，4组，2位索引 一组4个 3次

8路，16块 2组，1位索引，一组8个 不会替换

12. int32_t 大小4B，一个块存4个数据

A: 16个块，块16B，2路相联

∴ 8个组，块内偏移4位，索引3位

000	0000 (0)
000	0100 (1)
000	1000
000	1100
001	0000
⋮	⋮
111	1100 (31)
000	
⋮	
111	(63)
	(95)

实际：索引对应4数据

即1块，

同一索引可以对应3块（0~31，32~63，64~95）

缺失率 $\frac{8\times2}{64}=\frac{1}{4}$

B 16块，块16B 直接

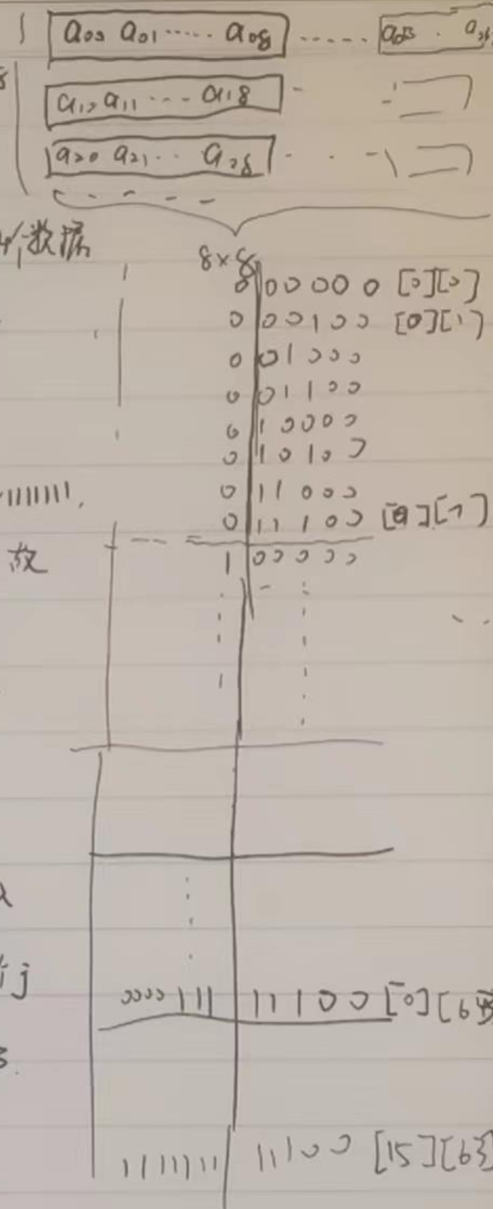
16组，索引4位 块内偏移4位

0000	0000 (0)
0000	0100 (1)
0000	1000 (2)
0000	1100 (3)
0001	0000
⋮	⋮
1111	1100 (63)
	⋮
	1

用索引可对应4块（0~63，16~95）

缺失率 $\frac{16}{64}=\frac{1}{4}$

13. 两层循环对调即可



14. (1) 4KB ÷ 32B = 128块

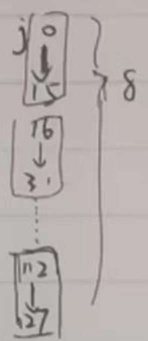
1个块32B可存8个数据，共能存1024个数据

数组共8192个数据，1024块

直接映射，索引7位

优化后：缺失率 $\frac{1}{8}$，共1024次

优化前：



索引位为0000000~1111111，即映射了16个j，故会替换，全部缺失8192次。

(2) 优化后：1024次（同(1)）

优化前：缺失 $\frac{1024}{8192} \times 8192 = 1024$次

(3) 优化前需要索引扩充至能存下128个j

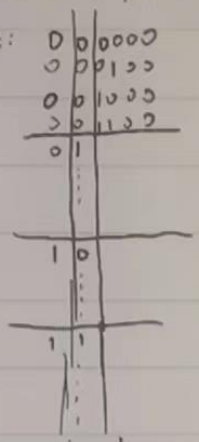
故需扩至 $\frac{8192}{8} \times 32B = 32KB$

优化后无需扩大，仍为4KB

15. 缓存32B. 块大小16B.

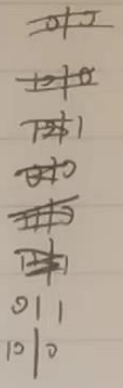
共2块. 索引1位, 块内偏移4位.

input:



output:

缓存

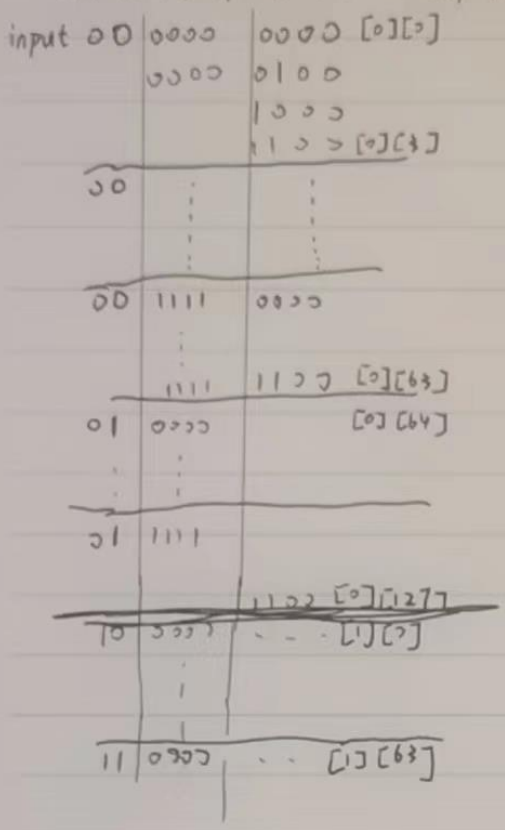


input					output			
列0	1	2	3	0	1	2	3
行0	miss	hit	miss	hit	miss	miss	m	m
1	miss	hit	miss	hit	miss	miss	m	m
2	m	h	m	h	miss	miss	m	m
3	m	h	m	h	miss	m	m	m

16. (1) 512 B, 2路, 块16 B.

共32块, 16组

索引4位, 块内偏移4位, 块标记4位, 命中率 $\frac{3}{4}$



(2) 不能. 存在缺失是由于随组号推进索引发生变化, 增加总大小仅改变块, 组数, 索引该表达限定

(3) 可以. 增块大小, 使得一个组、块中可存数据量变大, 组数、块数减小, 索引位可以减少. 使得部分原先索引位变化的地方现在索引位不会改变, 避免缺失.