

4.

固态内存保护机制与页表中表示的权限

- (1) PMP 中 X/W/R 作为保护机制，相当于一个三次保险：即使页表中表示有某一枚权限，但 PMP 中表示没有该权限，硬件仍然无法执行该操作。
- (2) L：表示是否锁定了 PMP 和对应的地址寄存器，锁定后则无法更改。
A：表示了是否启用此 PMP

$$(1) \frac{4 \times 1000}{8} = 500$$

固态内存保护机制与页表

$$\therefore \frac{2^{64}}{50} \times 4 + \frac{1}{1000} \div \frac{1}{1000} + \frac{1}{1000} \div \frac{1}{1000} = 147574 \text{ PB}$$

$$(2) \frac{2^{64}}{50} \times 4 \div \frac{1}{1000} \div \frac{1}{1000} \div \frac{1}{1000} = 2251.8 \text{ GB}$$

所需空间被降到 2251.8 GB

- (3) 因为多级页表只需将必要的页表加载到内存中，省去了大量未映射的页表项，所以大大减少了存储空间。



扫描全能王 创建

6.

因为地址的高位通常区别较小，会导致多个地址被分配到组中的同一位置，增大了竞争的可能，不仅无法较好的利用全部缓存空间，还会增大缓存判断数据竞争的开销，降低性能。

7.

这样可以使缓存的地址映射与虚拟内存系统的地址映射保持一致，不仅可以使用同一套硬件，减少硬件开发开销，还可以使虚拟地址和缓存互通，相互进行地址转换时更加方便。

f.

$$(1) \bar{t} = (1 \times 97\% + 110 \times 3\%) \times T_{cycle} = 4.27 T_{cycle}$$

$$= 0.1 \times 4.27 + 0.9 \times (4.27 - 1) = 3.7$$

$$(2). 命中率 = \frac{64KB}{1GB} = \frac{1}{16384}$$

$$\therefore \bar{t} = \left[\frac{1}{16384} \times 1 + 110 \times \left(1 - \frac{1}{16384}\right) \right] T_{cycle} \approx 110 T_{cycle}$$

(3). 因为内存访问有空间和时间上的局部性，导致了程序并不是随机地访问数据，所以在缓存容量很小的情况下也会有较高的命中率，而不是完全随机时约等于0的命中率。所以在高命中率时即可有效减少访问时间。

$$(4) 1 \times \alpha\% + 110 \times (1 - \alpha\%) \leq 105$$

$$\text{解得 } \alpha\% > 4.59\%.$$



9.

信号 机位板 侵入大小 块大小 拆装度 组装量 组装位板 标签纸板 倾斜度

1 32 4 64 8 8 16 32 16 32 16 32 16 16 16

2 32 4 64 8 8 16 32 16 32 16 32 16 16 16

3 32 4 64 16 1 0 26 6 6

4 32 16 64 1 256 8 18 6

5 32 16 128 2 64 6 19 7

6 32 64 16 256 8 18 16 16

7 32 64 16 64 16 64 16 16 16

8 32 64 128 16 32 16 16 16

10.

$$(1) t_A = (1-p_1) \times 0.22 + p_1 \times 100 = 0.77 \times (1 - 0.3) + 0.3 \times 100 = 77$$

$$t_B = (1-p_2) \times 0.52 + p_2 \times 100$$

$$t_A < t_B \quad \frac{1}{0.77} = \frac{1}{0.52} = 1.52 \quad 1.52 < 1$$

$$\text{解得 } 0.9978 p_1 < 0.3 + 99.35 p_2 \quad 0.9978 p_1 < 0.3 + 99.35 p_2$$

$$(2) t_A = (1-p_1) \times 0.22 + p_1 \times k \times 0.52$$

$$t_B = (1-p_2) \times 0.52 + p_2 \times k \times 0.52$$

$$t_A < t_B \quad 0.9978 p_1 < 0.3 + 52 p_2$$

$$\text{解得 } 22 p_1 < \frac{32}{k-1} + 52 p_2$$

$$201 > (52 - 1) \times 0.1 + 52 \times 1 \quad 201 > 51 + 52 \times 1 \quad 201 > 103$$

解得 $k < 5.2$ 最小



11.

列出二进制地址

0x000: 000 000 000 100 100 000 000 000

0x100: 000 000 000 000 100 100 000 000

0x101: 000 000 000 000 000 100 100 000

0x102: 000 000 000 000 000 000 100 100

0x103: 000 000 000 000 000 000 010 100

0x104: 000 000 000 000 000 000 000 100

0x105: 000 000 000 000 000 000 000 010

① 直接映射： $index = 4$ 位

∴ 替换 5 次

② 2 路： $index = 3$ 位， 有 2 个选择

∴ 替换 3 次

③ 4 路： $index = 2$ 位， 有 4 个选择

∴ 替换 1 次

④ 8 路： $index = 1$ 位

∴ 替换 0 次



扫描全能王 创建

12.

$$\text{块数} = \frac{256}{16} = 16$$

分析程序：会进行 96×100 次循环，每次都会读取 array 首地址，且根据 LRU 优化原则，每次都会将 i, j 有入栈中，所以需要读取 i, j

① B：直接映射

每次 j 都不同，并且 $96 > 16$ ，所以每次 j 都会 miss。

首地址永远不会变，在 96 个循环内相同

→ 第一次：3 个 miss

第二次：1 个 miss

：

第十四次：1 个 miss。缓存已满。

第十五次：1 个 miss。array 被替换

第十六次：array miss；被替换为 array， i 也 miss， j miss，三次

第十七次：1 个 miss

：

第九十六次：与十六次相同，三次 miss。 $j=0$

第九十七次：array hit，但 i miss，二次 miss

：

第 109 次：一个 miss。array 被替换。

第 110 次：array miss。但 array 放在了 $j=0$ 处， $i=1$ 仍 hit，2 次 miss

第 111 次：array hit。 i hit。但读新 j 替换，一次 miss

第 112 次：array hit。 i miss。 j miss。2 次 miss。

：

第 192 次：与 112 次相同

第 193 次：array hit。 i miss。放入 $i=2$ ，两次 miss。

：



扫描全能王 创建

所以：整个过程可看作 50 次循环。

第一次 miss 219 次 (包含初始化 array)。

剩下 49 次 miss 218 次

共 miss 10901 次。

共访问缓存 $3 \times 96 \times 100$ 次

$\therefore B$ 的缺失率为 37.85%，即叶所占的磁盘空间

固然是小部分，但固态硬盘的读写速度

远远大于机械硬盘

固态硬盘

固态硬盘

固态硬盘

固态硬盘

综上所述，固态硬盘比机械硬盘更省电，寿命更长，稳定性更好。

固态硬盘

固态硬盘

固态硬盘

固态硬盘

综上所述，固态硬盘比机械硬盘更省电，寿命更长，稳定性更好。

综上所述，固态硬盘比机械硬盘更省电，寿命更长，稳定性更好。

综上所述，固态硬盘比机械硬盘更省电，寿命更长，稳定性更好。

固态硬盘

综上所述，固态硬盘比机械硬盘更省电，寿命更长，稳定性更好。



扫描全能王 创建

13.

修改为

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

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

```
A[j][i] = A[j][i] + 1;  
} {  
    j++;  
    if (j == 128) {  
        i++;  
        j = 0;  
    }  
}
```

14. 块数 = $\frac{4 \times 1024}{32} = 128$

一个 int 4 byte

一个块中可以存 4 个 int

(1) 优化前. 在 j 的 128 次循环中. 每次会 miss 1 次. 共 $128 \times 1 = 128$ 次

共 $32 \times 64 = 2048$ 次

优化后. 在 j 的 128 次循环中. 每次 miss 1 次. 共 $128 \times 1 = 128$ 次

共 miss 32 次

(2). 优化前. 在 i 的 64 次循环中. 每 4 次有一次 128 个 j 全 miss

共 $128 \times 16 = 2048$ 次

优化后. 在 i 的 64 次循环中. i 每 4 次 miss 1 次.

共 $128 \times 16 = 2048$ 次.

(3). 优化前 优化后 都需要. $128 \times 64 \times 4 \div 1024 = 32$ kB



扫描全能王 创建

15.

Input

3| 0 3| 1 3| 2 3| 3

out per

3| 0 3| 1 3| 2 3| 3

行0	miss	hit	hit	hit	miss	miss	miss	miss
行1	hit	hit	hit	hit	hit	hit	hit	hit
行2	miss	hit	hit	hit	miss	miss	miss	miss
行3	hit	hit	hit	hit	hit	hit	hit	hit

$$16. \text{快取} = \frac{512}{16} = 32, \text{一个块4个int}$$

$$80 = \frac{128}{32} = 4 \text{块}$$

(1) 在 ~~128 × 2~~ 次访问中，只会 miss 2 次

命中率为 99.22%

(2) 可以，因为会增加每一路中的快取量，命中率会增加。而总的当一路中的快取多过 32 块时，则命中率为 100%。

(3) 不会，因为总大小不变时，每一路同样只能有 64 个 int，所以命中率不会变。

命中率 = $\frac{\text{命中次数}}{\text{总访问次数}} \times 100\%$

$$\text{命中率} = \frac{1}{4} \times 100\% = 25\%$$

命中率 = $\frac{\text{命中次数}}{\text{总访问次数}} \times 100\%$

$$\text{命中率} = \frac{3}{4} \times 100\% = 75\%$$

命中率 = $\frac{\text{命中次数}}{\text{总访问次数}} \times 100\%$ ，命中率与访问的顺序无关。

: 7.7.5



扫描全能王 创建