1. **（书中习题4第15题）以下是对矩阵进行转置的程序段：**

**typedef int array[4][4];**

**void transpose(array dst, array src)**

**{**

**int i, j;**

**for (i = 0; i < 4; i++)**

**for (j = 0; j < 4; j++)**

**dst[j][i] = src[i][j];**

**}**

**假设该段程序运行的计算机中sizeof(int)=4，且只有一级cache，其中L1 data cache的数据区大小为64B，采用直接映射、写回方式，块大小为16B，初始为空。数组dst从地址0000C000H开始存放，数组src从地址0000C040H开始存放。填写下表，说明数组元素src[row][col]和dst[row][col]映射到cache的哪一行，其访问是命中（hit）还是失效（miss）。**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **src数组** | | | | **dst数组** | | | |
| **64B** | **col=0** | **col=1** | **col=2** | **col=3** | **col=0** | **col=1** | **col=2** | **col=3** |
| **row=0** | **Miss** | **miss** | **hit** | **hit** | **miss** | **miss** | **Hit** | **Hit** |
| **row=1** | **Miss** | **hit** | **miss** | **hit** | **miss** | **Miss** | **Miss** | **Hit** |
| **row=2** | **Miss** | ***hit*** | **hit** | **miss** | **miss** | **Hit** | **Miss** | **miss** |
| **row=3** | **Miss** | **hit** | **hit** | **hit** | **miss** | **Hit** | **Hit** | **miss** |

**参考答案：**

**数据区大小32B或128B的情况：**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **src数组** | | | | **dst数组** | | | |
| **64B** | **col=0** | **col=1** | **col=2** | **col=3** | **col=0** | **col=1** | **col=2** | **col=3** |
| **row=0** | **0/miss** | **0/miss** | **0/hit** | **0/hit** | **0/miss** | **0/miss** | **0/hit** | **0/hit** |
| **row=1** | **1/miss** | **1/hit** | **1/miss** | **1/hit** | **1/miss** | **1/miss** | **1/miss** | **1/hit** |
| **row=2** | **2/miss** | **2/hit** | **2/hit** | **2/miss** | **2/miss** | **2/hit** | **2/miss** | **0/miss** |
| **row=3** | **3/miss** | **3/hit** | **3/hit** | **3/hit** | **3/miss** | **3/hit** | **3/hit** | **1/miss** |

**参考答案：**

**从程序来看，数组访问过程如下：**

**src[0] [0]、dst[0] [0]、src[0] [1]、dst[1] [0]、src[0] [2]、dst[2] [0]、src[0] [3]、dst[3] [0]**

**src[1] [0]、dst[0] [1]、src[1] [1]、dst[1] [1]、src[1] [2]、dst[2] [1]、src[1] [3]、dst[3] [1]**

**src[2] [0]、dst[0] [2]、src[2] [1]、dst[1] [2]、src[2] [2]、dst[2] [2]、src[2] [3]、dst[3] [2]**

**src[3] [0]、dst[0] [3]、src[3] [1]、dst[1] [3]、src[3] [2]、dst[2] [3]、src[3] [3]、dst[3] [3]**

**因为块大小为16B，每个数组元素有4个字节，所以4个数组元素占一个主存块，因此每次总是调入4个数组元素到cache的一行。**

**当数据区容量为64B时，L1 data cache中共有4行。数组元素dst[0][i] 、src[0][i] (i=0～3)都映射到cache第0行；数组元素dst[1][i] 、src[1][i] (i=0～3)都映射到cache第1行，……，即dst[j][i]、src[j][i] (j, i=0～3)都映射到cache第i行。因此，从上述访问过程来看，src[0][0]所在的一个主存块（即存放src[0][i] (i=0～3)四个数组元素）刚调入cache后，dst[0][0]所在主存块又把src[0][0]替换掉了。……**

1. **在书中习题4第24题题干给出的条件下，假定该机采用小端方式， x为short型变量，其虚拟地址为02EAH，则x能否在cache中命中，若能，则x为多少？若不能，为什么？若x的虚拟地址为0E58H，则结果如何？（要求写出过程）**

**02EAH=0000 0010 1110 1010B（其中黄色为页内地址，蓝色为TLB组号），其虚页号为0000001 01B，根据TLB中第1组的情况，知在TLB中缺失，**

**然后重新按虚页号0000001 01B找到页表中第005H项，知其有效，并得到物理页号为16H=10110B，和页内地址拼接成物理地址为10110 1101010B。**

**以上物理地址即主存地址，按cache工作机制重新划分为三段101101 1010 10（其中黄色为块内地址，蓝色为cache行号，最前面为标记），cache行索引为1010B，找到第A行，知有效并标记为2DH=101101B，与x地址高位相同，故在cache中命中。取出“第2”和“第3”字节。因为是小端方式，故其数值为43 62H。**

**类似的，0E58H=0000 1110 0101 1000B，其虚页号为0000111 00B，根据TLB中第0组的情况，知在TLB中命中，得到物理页号为10H=10000B，和页内地址拼接成物理地址为10000 1011000B，主存地址划分为100001 0110 00，cache行索引为0110B，找到第6行，知有效位为0，故在cache中不命中。**

1. **主存按字节编址，数组在内存按行优先方式存放。Cache数据区为1024B，块大小为16B，直接映射，gd的地址从0开始，sizeof(int)=4。求以下代码段A、B和C对数组gd的访问中，各自共有多少次读操作？三段代码各有多少次不命中？若Cache数据区扩大为2048B，则各有多少次不命中？**

**struct algae\_pos{**

**int x;**

**int y;**

**} ;**

**struct algae\_pos gd [16][16];**

**int total\_x=0, total\_y=0;**

**int i, j;**

**…**

**for (i = 0; i < 16; i++) {**

**for (j = 0; j<16; j++) {**

**total\_x+= gd[i][j].x;**

**total\_y+= gd[i][j].y;**

**}**

**}**

**…**

**struct algae\_pos{**

**int x;**

**int y;**

**} ;**

**struct algae\_pos gd [16][16];**

**int total\_x=0, total\_y=0;**

**int i, j;**

**…**

**for (i = 0; i < 16; i++) {**

**for (j = 0; j<16; j++) {**

**total\_x+= gd[i][j].x;**

**}**

**}**

**for (i = 0; i < 16; i++) {**

**for (j = 0; j<16; j++) {**

**total\_y+= gd[i][j].y;**

**}**

**}**

**…**

**struct algae\_pos{**

**int x;**

**int y;**

**} ;**

**struct algae\_pos gd [16][16];**

**int total\_x=0, total\_y=0;**

**int i, j;**

**…**

**for (i = 0; i < 16; i++) {**

**for (j = 0; j<16; j++) {**

**total\_x+= gd[j][i].x;**

**total\_y+= gd[j][i].y;**

**}**

**}**

**…**

**代码段A 代码段B 代码段C**

**三段代码读数组gd中元素的次数都是16×16×2=512次。**

**数组gd总大小为16×16×2×4=2048B。每两个数组元素的大小为2×2×4=16B，故两个元素为一个主存块。对于cache大小为1024B数据区的情况，数组大小是cache数据区的两倍，所以，gd[0][0]～gd[7][15]之间的数据与gd[8][0]～gd[15][15]之间的数据在cache中将相互覆盖替换，因此，三段代码的访问情况如下：**

**对于代码段A，每块的连续两次访问（gd[i][j].x和gd[i][j+1].x或gd[i][j].y和gd[i][j+1].y）中，总是第一次缺失，第二次命中，故缺失次数为256。（一拖一）**

**对于代码段B，每块的连续两次访问（gd[i][j].x和gd[i][j].y或gd[i][j+1].x和gd[i][j+1].y）中，总是第一次缺失，第二次命中，故缺失次数为256。（一拖一）**

**对于代码段C，每块连续4次访问（gd[i][j].x、gd[i][j].y、gd[i][j+1].x、gd[i][j+1].y）中，总是第一次缺失，后面三次命中，故缺失次数为128。（一拖三）**

**对于2048B数据区的情况，数组正好能够全部装入到cache，不会相互替换覆盖。所以，在2048B情况下数组元素不会被替换出来，因此，对于三段代码，每块中的4次访问，都是第一次缺失，以后三次命中，故缺失次数都为128。（一拖三）**