

# Homework: 存储

本此作业统一以  $K = 10^3$ ,  $M = 10^6$ ,  $G = 10^9$  为计量单位。以后如果遇到类似的题目请提前查看/询问/确定这件事。

请直接用 Markdown 题目源文件填充答案，最后统一提交 PDF 格式，比如使用 Typora 导出。

## T1

以下是A型号磁盘的相关参数

参数	值
盘片数	2
每个盘片的面数	2
柱面数（也叫磁道数）	500000
平均每条磁道的扇区数	500
扇区大小	4096 Byte
旋转速率	5400 RPM
平均寻道时间	10ms

### 1.1

求该磁盘的容量（GB 为单位）。

#### 1.2.1

求该磁盘访问一个扇区的平均延迟（ms 为单位）。

#### 1.2.2

求该磁盘随机读写时的每秒访问次数（IOPS）。

提示：思考磁盘的最小访问单位是什么，书上有提到

#### 1.2.3

求该磁盘的顺序读取带宽（MB/s 为单位）。

其中，顺序读取带宽的定义是

$$\max\left\{\forall \text{可以存储在磁盘 } A \text{ 上的文件 } F \frac{F \text{ 的大小}}{\text{磁盘从随机时刻开始，顺序读取完 } F \text{ 所需的期望时间}}\right\}$$

为了答案统一，有如下假设：

1. 不能确认为 0 的值，都认为是以平均值为期望的随机数（比如即使是顺序地访问磁道，每次寻道时间也是以平均寻道时间为期望的随机数）

2. 顺序存储的文件在相邻的扇区上是连续，当大小超过一个盘面的一个磁道可以容纳的空间时，你可以自己决定下一个开始存储的位置。显然本题你需要想一想什么存法读取时更快。

可以在答案中附上你答案对应的文件的存储方式（可以画图）

## T2

下面的表给出了一些不同的高速缓存的参数。你的任务是填写出表中缺失的字段。其中 m 是物理地址的位数，C 是高速缓存大小（数据字节数），B 是以字节为单位的块大小，E 是相联度，S 是高速缓存组数，t 是标记位数，s 是组索引位数，而 b 是块偏移位数。

m	C	B	E	S	t	s	b
32		4	4		24	6	
32	1024	32	2				
32	2048			256	21	8	3

## T3

假设我们有一个具有如下属性的系统：

- 内存是字节寻址的。
- 内存访问是对 1 字节字的（比如访问一个 char）。
- 地址宽 12 位。
- 高速缓存是两路组相联的(E=2)，块大小为 4 字节(B=4)，有 4 个组(S=4)。

高速缓存的内容如下，所有的地址、标记和值都以十六进制表示：

组索引	标记	有效位	字节1	字节2	字节3	字节4
0	00	1	40	41	42	43
	83	1	FE	97	CC	D0
1	00	1	44	45	46	47
	83	0	54	55	56	57
2	00	1	48	49	4A	4B
	40	0	21	22	23	24
3	FF	1	9A	D0	03	EE
	00	0	A1	A2	A3	A4

对于下面每个内存访问，当他们顺序执行时，指出高速缓存是否命中，如果命中且操作前的数可从已有信息判断，请给出。

操作	地址	命中	值（或未知）
读	0x834		
写	0x836		
读	0xFFF		

# T4

仔细阅读下面的程序，根据条件回答下列各题：

- 地址宽度为 10
- 数组的起始地址为 0b0001000000（即二进制表示）
- Block size = 4 Byte，Set = 4，两路组相连（B = 4, S = 4, E = 2）
- 替换算法为 LRU (最近最少使用)

```
#define LENGTH 8
void clear44(char array[LENGTH][LENGTH]) {
    int i, j;
    for (i = 0; i < 4; i++)
        for (j = 0; j < 4; j++)
            array[i][j] = 0;
}
```

## 4.1.1

以上程序会发生几次缓存miss？

## 4.1.2

如果 LENGTH = 16，那么会发生几次缓存miss？

## 4.1.3

如果 LENGTH = 17，那么会发生几次缓存miss？

## 4.1.4

请画出在 LENGTH = 17 时，程序执行结束时 set0 和 set1 的高速缓存状态，假设一开始全空。

可以用 `Array[0][0] ~ Array[0][3]` 的形式表示 Data 段落，有效位为 0 的行留空，每个 Set 内的顺序无所谓

SetID	Tag	Data
0		
0		
1		

SetID	Tag	Data
1		

4.2

修改条件为

- 地址宽度为 10
- 数组的起始地址为 0b0010000000（即二进制表示）
- Cache 的容量为 16 Byte，Block size = 4 Byte，全相联
- 替换算法为 LRU

4.2.0

Tag 的位数是多少？

4.2.1

原始程序会发生几次缓存miss？

4.2.2

如果 LENGTH = 16，那么会发生几次缓存miss？

4.2.3

如果 LENGTH = 17，那么会发生几次缓存miss？

4.2.4

请画出在 LENGTH = 17 时，程序执行结束时的高速缓存状态，假设一开始全空。

可以用 `Array[0][0] ~ Array[0][3]` 的形式表示 Data 段落，有效位为 0 的行留空

SetID	Tag	Data
0		
1		
2		
3		