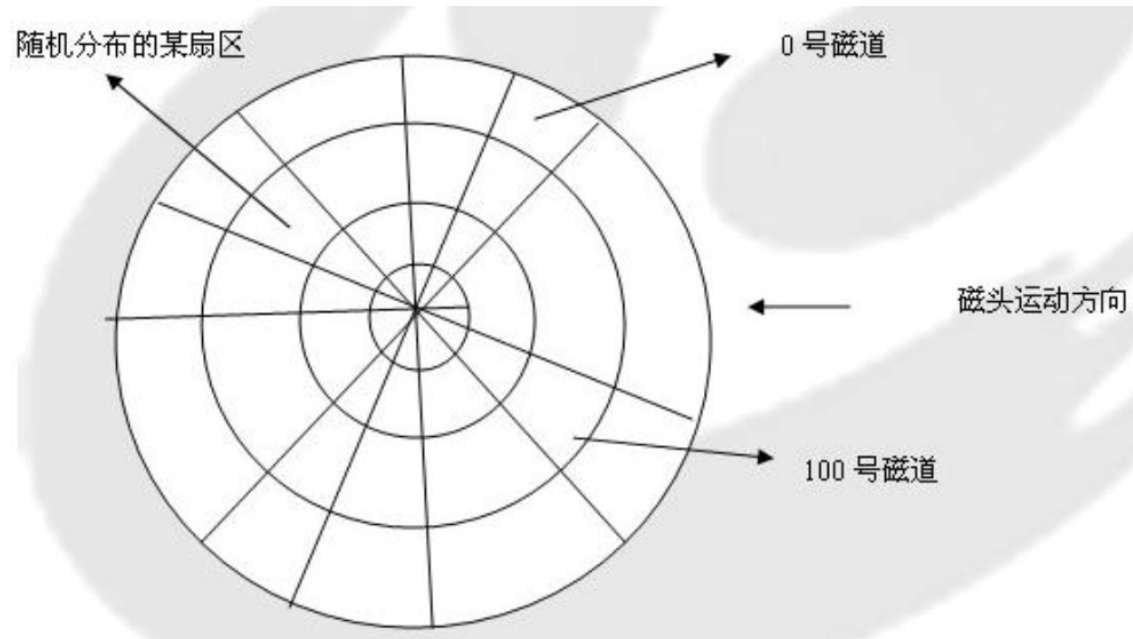


Homework8

10

问题 1

- 计算采用 C-SCAN 磁盘调度策略，使用 2KB 的内存空间记录 16384 个磁盘块的空闲状态。
 - (1) 说明在上述条件下，如何进行磁盘块空闲状态管理
 - (2) 某单面磁盘转速为 6000 RPM，每个磁道有 100 个扇区，相近磁道的平均移动时间为 1ms。当前磁头位于 100 号磁道处，并沿磁道号较大的方向移动。磁道号请求队列为 50、90、30、120，每个请求需要读取一个随机分布的扇区，读完这 4 个扇区需要多少时间？
 - (3) 如果磁盘替换随机访问的 Flash 设备，什么方法比 C-SCAN 更有效？



问题 2

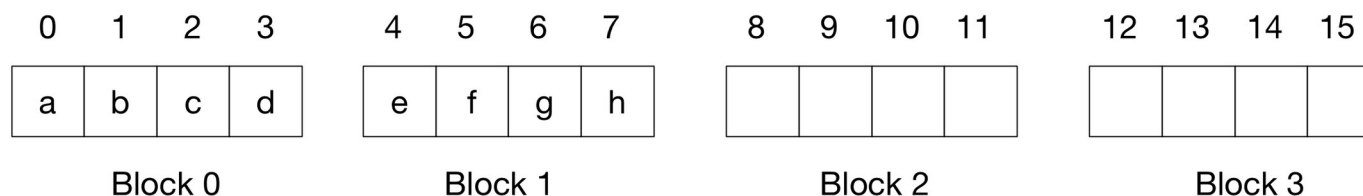
- SSD FTL 采用 hybrid mapping 的方法管理，规定 log table 中最多有 3 条记录，初试状态如下图所示。假设更新 page 都是连续写入一个 block 的，不论更新 page 是来自于哪个 block。
- 用户依次对以下数据块进行更新操作：

e, f, a, a, a, g, g, f, h, e

- 求完成上述请求序列的过程中：1) 写放大为多少个 page？ 2) 一共发生了几次擦除操作？ 3) 各发生了几次 switch, partial merge 和 full merge？ 4) 画出上述请求需求完成后的系统状态图（包括 log table, data table 和闪存中的数据块），空白 block 的分配策略采用 first-fit

Log Table:

Block Table: 1000->0, 2000->4



问题 3

- 文件 foo.txt 的内容是字符串 “foofoofoo\n”，文件 bar.txt 的内容是字符串 “barbarbar\n”
- 执行右侧程序，请写出所有可能的输出。

```
#include "csapp.h"
int main() {
    int fd1, fd2;
    char c;
    fd1 = open("foo.txt", O_RDONLY, 0);
    fd2 = open("bar.txt", O_RDONLY, 0);
    close(fd2);
    fd2 = open("bar.txt", O_RDONLY, 0);
    int rc = fork();
    if (rc == 0) {
        read(fd2, &c, 1);
        printf("fd2=%d, child: c=%c\n", fd2, c);
    }
    read(fd2, &c, 1);
    printf("fd2=%d, c=%c\n", fd2, c);
    exit(0);
}
```