

FS作业2

1.假设文件系统采用 FFS，块大小为 4KB。执行下面代码:

```
#define MAX (1000)
char buf[MAX];
int fd = open("tmpfile", O_CREAT | O_TRUNC | O_RDWR, 0666); int n=0,
i=0;
if (fd < 0 ) {
    perror("open");
    exit(-1);
}
for (i = 0; i < MAX; i++) {
    bzero(buf, sizeof(buf));
    sprintf(buf, "%3d\n", i);
    n = write(fd, buf, strlen(buf)); printf("len=%d\n", strlen(buf));
    if (n != strlen(buf)) {
        perror("write");
        printf("length=%d, buf=[%s]", strlen(buf), buf);
    }
}

close(fd);
```

- 1) 请问这段代码执行成功后，tmpfile 的大小是多少字节？
- 2) 在没有文件缓存的情况下，总共写几次磁盘，每次多少字节？
- 3) 在有文件缓存的情况下，总共写几次磁盘，每次写多少字节？

2. 在有文件缓存的情况下，在一个 FFS 文件系统中创建一个文件“/home/OS18/fs03.pdf”，需要写几个块？写哪几个块？如果在任意时刻发生宕机，会出现哪些不一致？请详细列出所有不一致的情况。

3. 磁盘上有一个长度为 20KB 的文件 A，如果一个进程打开文件 A，并调用 write 一次性向文件 A 的块 0 和块 1 写入新数据。假设有文件缓存，宕机可能发生在任意时刻。

1) 如果文件系统采用数据日志，宕机恢复后，文件 A 的内容是什么？分不同情况讨论(在什么情况下，文件 A 的内容是什么)。

2) 如果文件系统采用元数据日志，并且采用一致性修改，宕机恢复后，文件 A 的内容是什么？分不同情况讨论(在什么情况下，文件 A 的内容是什么)。

4. LFS 的 imap 采用类似数组的结构，下标是 ino，每项保存 i-node 的磁盘地址，例如，imap[k]记录 ino 为 k 的 i-node 的磁盘地址。假设一个 LFS 的块大小为 4KB，磁盘地址占 4 字节。如果已经分配了 500 万个 i-node，请问

1) 它的 imap 有多少个块？给出计算过程

2) 它的 CR 有多少个块？给出计算过程

3) 如何查 ino=654321 的 inode 的磁盘地址。给出查找和计算过程。

5. 一个 LFS 的块大小为 4KB，segment 大小是 4MB。文件块采用与 FFS 一样的多级索引，每个指向数据块的指针占 4 字节。该 LFS 中已经有一个 20MB 的文件 foo，

1) 给出文件 foo 的文件块索引结构

2) 写文件 foo 的第 2560 块(假设它在磁盘块 A_i 中， A_i 为磁盘逻辑块号)，需要写哪些块？给出它们写在磁盘上的顺序。需要几次 I/O？

3) 如果是 FFS (其块大小也为 4KB)，写文件 foo 的第 2560 块，需要写哪些块？需要几次 I/O？

4) 如果是日志文件系统，只记录元数据日志，且日志不采用批量提交，则写文件 foo 的第 2560 块，需要写哪些块？需要几次 I/O？