

## 作业 14

毕定钧 2021K8009906014

本次作业包含:

14.1 现有一个文件系统，它的文件块索引采用多级间址。该文件系统的 *inode*，包含 10 个直接指针，1 个一级间址指针，1 个二级间址指针和 1 个三级间址指针。假设文件块大小为  $4KB$ ，每个文件块对应的磁盘块地址为  $4B$ 。取  $1K = 1024$ 。

(1) 请问该索引结构能够索引的最大文件是多大？

(2) 请问一个  $1GB$  的文件需要几级间址？它总共有多少间址块？其中，各级间址块分别是多少？如何找到第 20,000 块？

14.2 某用户  $X$  刚挂载了一个文件系统（假设此时该文件系统的所有 *inode* 已被加载到内存），该文件系统使用的磁盘块大小为  $4KB$ ，能用到的 *page cache* 大小最大为  $512MB$ 。随后，该用户执行如下所示程序  $A$ 。请分析（请写出分析过程）

(1) 当程序  $A$  打开 *fs02.ppt* 文件时，文件系统需要从磁盘读取几个磁盘块？

(2) 假设该文件系统采用 *write through* 的缓存策略，当程序  $A$  完成对 *fs02.ppt* 的写入操作后，文件系统写几次磁盘块？分别写哪些磁盘块？如果该文件系统采用的是 *write back* 缓存策略，那么程序  $A$  在写完 *fs02.ppt* 还未关闭文件时，文件系统共写入了几个磁盘块？

(3) 程序  $A$  执行完成后，用户  $Y$  再次运行该程序，当程序  $A$  打开 *fs02.ppt* 时，文件系统需要从磁盘读取几个磁盘块？

(4) 用户  $Y$  将程序  $A$  中打开的文件修改为 */home/os23/fs01.ppt*，并编译执行程序  $A$ ，那么当程序  $A$  打开 *fs01.ppt* 时，文件系统需要从磁盘读取几个磁盘块？

注：假设 (1) 每个目录下的所有条目都只占用 1 个磁盘块；(2) *fs01.ppt* 和 *fs02.ppt* 两个文件已在文件系统中存在，且 *fs02.ppt* 的文件长度超过  $1MB$ 。程序  $A$  代码如下

```
#define MAX (1024)
char buf[MAX];
int fd = open("/home/os23/fs02.ppt", O_CREAT | 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, "%6d\n\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);
```

14.3 现有一个文件系统，在其使用文件缓存的情况下，某个应用首先创建了一个文件 */home/os23/*

*fs03.pdf*”，再向该文件中写入了 4 KB 的数据，请分析该过程需要写几次磁盘块？分别写哪些块？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。

(注：假设 *home* 和 *OS23* 目录都已存在)

## 14.1

### (1)

每个索引块包含  $\frac{4KB}{4B} = 1K = 2^{10}$  个地址，故每个直接指针可以索引 4KB，每个一级间址指针能够索引  $2^{10}$  个直接指针即  $2^{10} \times 4KB = 4MB$ ，每个二级间址指针能够索引  $2^{10}$  个一级间址指针即  $2^{10} \times 4MB = 4GB$ ，每个三级间址指针能够索引  $2^{10}$  个二级间址指针即  $2^{10} \times 4GB = 4TB$ ，故这个索引结构能够索引的最大文件大小是  $4KB \times 10 + 4MB \times 1 + 4GB \times 1 + 4TB \times 1 = 40KB + 4096KB + 4194304KB + 4294967296KB = 4299165736KB$ 。

### (2)

需要二级间址，它需要 257 个间址块，其中包含 1 个二级间址块和 256 个一级间址块。为了找到第 20000 个数据块，应先计算  $\frac{20000}{1024}$  取商 19 和余数 544，那么就应从二级间址块中找到第  $19 + 1 = 20$  个一级间址指针，然后找到一级间址块并取出第 544 个直接指针，从而找到第 20000 个数据块。

## 14.2

### (1)

打开文件 *fs02.ppt* 时，需要先读取根目录对应磁盘块，之后读取 *home* 文件夹对应的磁盘块，然后读取 *os23* 文件夹对应磁盘块并找到文件 *fs02.ppt*。共需要读取 3 个磁盘块。

### (2)

在程序 A 完成写入后，文件系统会向磁盘写入 1 次，写入的是第 256 块的最后 1KB。*write through* 策略保证每次调用 *write* 函数均会立即执行一次写入磁盘，故程序 A 写入完成后仅会向磁盘写入最后一次程序 A 写入的内容。如果采用 *write back* 策略，那么在不发生块替换、写入未达到固定时间间隔的情况下，在文件关闭前不会向磁盘写入。

### (3)

程序执行完成并不会主动卸载磁盘块，所以之前载入的磁盘块依然在缓存中，再次执行无需读取磁盘块。

### (4)

同样的，此时文件 *fs01.ppt* 所在的文件夹 *os23* 对应磁盘块已在缓存中，无需读取磁盘块。

## 14.3

首先需要创建文件 *fs03.pdf* 的 *inode* 并更新位图，此时需修改两块，之后文件系统会更新目录结构，并修改文件夹 *os23* 的 *inode*，添加对 *fs03.pdf* 的索引，需要修改一块；然后为该文件分配一块用于存储数据，并更新 *fs03.pdf* 的 *inode* 使其能够索引数据块。

如果任意时刻宕机可能会发生：

元数据不一致：如果宕机发生在元数据更新之前，文件系统可能没有记录新文件的元数据，导致文件系统中不存在文件 `"/home/OS22/fs03.pdf"` 的信息。

部分数据写入：如果宕机发生在数据块的写入之前，可能只有部分数据被写入了磁盘，而文件系统的元数据已经被更新，文件大小等信息不一致。