操作系统 第十二次作业

姓名：朱首赫

学号：2023K8009906029

12.1 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 12 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。

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 5,000 块?

解： 1) 一个文件块可存储 4KB / 4B = 1024 个地址。直接索引 12 * 4KB = 48 KB，一级间址 1 * 1024 * 4KB = 4MB，二级间址 1 * 1024 * 1024 * 4KB = 4GB，三级间址 1 * 1024 * 1024 * 1024 *4KB = 4TB，因此改索引结构能索引的最大文件为 4TB + 4GB + 4MB + 48KB

2) 4MB + 48KB < 1GB < 4GB + 4MB，需要二级间址。

该文件共需要 $1GB/4KB = 2^{18}$ 个文件块，其中直接指针能提供 12 个文件块，一级间址能提供 1024 个文件块，需要二级间址提供 $2^{18} - 1024 - 12$ 个文件块，那么需要其指向的间址块中有 $\lceil (2^{18} - 1024 - 12) \div 1024 \rceil = 255$ ，每一项指向一个间址块。因此总共有 1 + 1 + 255 = 257 个间址块。其中二级间址块 1 个，一级间址块有 256 个。

5000 - 12 - 1024 = 3964，$\lfloor 3964 \div 1024 \rfloor = 3$，因此应该查找二级间址块的 3 号项指针（从 0 开始索引），然后查找其指向的一级间址块的第 3964%1024 = 892 项（即 891 号项），该项指针指向的数据块即为第 5000 个数据块。

12.2 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 256MB。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，文件系统写入几个磁盘块? 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，文件系统写入几个磁盘块?

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/os25/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）所有目录都只需 1 个磁盘块保存其内容；（2）fs01.ppt 和 fs02.ppt 两个文件已在文件系统中存在。

Listing 1: 程序 A 代码

```
1 #define MAX (1024)
```

```
char buf[MAX];
int fd = open("/home/os25/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, "%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);
```

解： 1) 由于所有 inode 都已被加载到内存，所以读 inode 不需要访问磁盘。路径解析：先找到 / 目录的 inode，读取根目录数据块，再找到 home 目录的 inde，读取其数据块，最后找到 os25 目录的 inode，读取其数据块，从而定位到 fs02.ppt 的 inode，完成打开操作。总共需要从磁盘读取 3 个磁盘块。

2) 程序 A 对 fs02.ppt 写入 1024 次 4B 的内容，共需要分配一个数据块。

若采取写穿透策略，首先分配数据块需要写一次 bit-map，然后每次写入 4B 的字符串时，都需要同时写入数据块和更新文件和父目录的 inode 元数据，因此共需要写入 1 + 1024 * 2 = 2049 个磁盘块。

若采取写回策略，由于还未关闭文件，写入内容全保留在内存中，写入 0 个磁盘块。

3) 0 个，因为内存中已有缓存。

4) 0 个，因为此时所有 inode 在内存中，且 / 、home 、os25 三个目录的数据块也已经在内存中，可以直接定位到 fs01.ppt 的 inode。