

操作系统 第十三次作业

姓名：朱首赫

学号：2023K8009906029

13.1 现有一个文件系统，在其使用 page cache 的情况下，某个应用创建了一个文件“/home-/OS25/fs03.pdf”，并往该文件中写入了 33 KB 的数据，请分析该过程需要写几个块？分别写哪几个块？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。（注：假设 home 和 OS25 目录都已存在）

解：(1) 创建和写入应该是两个独立的操作。创建操作：需要写 OS25 目录块、OS25 目录 inode、inode-bitmap、新文件 inode，共 4 个块。写入操作： $[33KB \div 4KB] = 9$ ，需要写新文件 inode、data-bitmap、9 个新数据块，共 11 个块。综上，创建和写入操作一共需要写入 15 个块。

(2) 假设采取写穿透策略。

创建文件：会产生四种 FS 不一致的情况：1. 若 inode-bitmap 写回，但新文件 inode 未写回，会产生垃圾 inode；2. 若新文件 inode 写回，但 inode-bitmap 未写回，可能造成重复分配。3. 若目录块写回，但新文件 inode 未写回，会产生悬空目录项。4. 若新文件 inode 写回，但目录项未写回，会产生孤立 inode。

写入文件：若 data-bitmap 写回，但数据块没写回，会造成垃圾块，即 FS 不一致。若文件 inode 写回，但数据块未写回，会导致数据与元数据不一致。

13.2 某个文件系统在磁盘上保存了一个大小为 20 KB 的文件 A，现有一个进程打开文件 A，并调用 write 函数一次性向文件 A 的文件块 0 和文件块 1 写入新数据。假设该文件系统使用文件缓存，且宕机可能发生在任意时刻。请分析

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？请分不同情况讨论（即在什么样的宕机情况下，文件 A 的内容是什么）；

2) 如果文件系统采用元数据日志，并且采用先改数据再改元数据的方式，宕机恢复后，文件 A 的内容是什么？请分不同情况讨论（即在什么样的宕机情况下，文件 A 的内容是什么）。

解：1) 若在写日志时宕机，恢复后文件 A 的内容为旧数据。若在日志提交后（包括 checkpoint 中）宕机，恢复后文件 A 的内容为新写入的数据。

2) 若在写入数据块时宕机，可能造成新旧数据混合的情况。若在写完数据块后，日志提交前宕机，文件 A 的内容为新写入的数据（假设块 0 和块 1 本来就存在），但文件元数据为旧值。若在日志提交后宕机，数据和元数据都是新值。

13.3 LFS 的 imap 和 CR 都采用类似数组的结构，下标是 ino 或 imap 块号，每一项保存对应 i-node 或 imap 块的磁盘地址。例如，imap[k] 记录 ino 为 k 的 i-node 的磁盘地址；CR[n] 记录第 n 个 imap 块的磁盘地址。假设一个 LFS 的块大小为 4KB，磁盘地址占 4B。如果已经分配了 100 万个 i-node，请问：

1) 该 LFS 的 imap 有多少个块？请给出计算过程；

2) 该 LFS 的 CR 有多少个块？请给出计算过程；

3) 如何查 $\text{ino}=356302$ 的 inode 的磁盘地址？请给出查找和计算过程。

解：1) 一个 imap 块可以存储 $4\text{KB}/4\text{B} = 1024$ 个 i-node 的磁盘地址，因此目前有 $[1000000/1024] = 977$ 块 imap。

2) 同上，一个 CR 块可以存储 1024 个 imap 的磁盘地址，足够容纳上述 997 块 imap。但 LFS 一般设置两个 CR 块作为交替快照。

3) imap 索引 $= 356302 / 1024 = 347$, 查找 CR[347] 得到 imap 块地址；inode 索引在 imap 块中的偏移 $= 356302 \% 1024 = 974$, 查找 CR[347][974] 即可得到 inode 的磁盘地址。

13.4 一个 LFS 的块大小为 4KB, segment 大小是 4MB。文件块采用多级索引，即包含 10 个直接指针，以及一、二、三级间接指针各 1 个。每个指向数据块的指针占 4 字节。该 LFS 中已经有一个 12MB 的文件 foo，请分析：

1) 给出文件 foo 的文件块索引结构，即文件 foo 使用了哪些指针？

2) 在该 LFS 中写文件 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？需要几次 I/O？请给出它们写在磁盘上的顺序；

3) 如果是 Fast FS (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？

解：1) 一个索引块能存储 $4\text{KB} / 4\text{B} = 1024$ 个地址。10 个直接索引管理的数据块大小为 $10 * 4\text{KB} = 40\text{KB}$ ，一级间接指针管理的数据块大小为 $1024 * 4\text{KB} = 4\text{MB}$ 。二级间接指针管理的数据块大小为 $1024 * 4\text{MB} = 4\text{GB}$ ，三级间接指针管理的数据块大小为 $1024 * 4\text{GB} = 4\text{TB}$ 。

文件 foo 包含 $12\text{MB} / 4\text{KB} = 3072$ 个数据块，其中直接索引管理 10 个数据块，一级间接索引管理 1024 个数据块，二级间接索引管理 $[3072 - 1024 - 10]/1024 = 2$ 个一级间接索引，进而管理 2038 个数据块。

2) LFS 采取异地更新，写数据块需要更新索引， $2560 > 10 + 1024$ ，需要写二级间接索引块、一级间接索引块、数据块、新的 inode、新的 imap，共 5 个块。LFS 将所有修改缓存在内存的 Segment 中，一次性顺序写入磁盘，所以只需要 1 次 I/O。

3) FSS 采取原地更新，只需要写数据块和 foo 文件的 inode，共 2 个块。不考虑路径解析的 I/O，考虑缓存因此不需要读出，假设采取写穿透策略，共需要 2 次 I/O。

4) 假设采用先改数据再改元数据的方式，且不批量提交，需要写日志块 (~~TxB+inode~~ 修改)、提交块 (TxE)、数据块、inode，共 4 个块 + 清除日志。在不宕机的情况下，共需要 4 次 I/O。**5-6**

5

1-1