第十一章 文件系统实现

1. What are the advantages of the variation of linked allocation that uses a FAT to chain together the blocks of a file?

2. Some file systems allow disk storage to be allocated at different levels of granularity. For instance, a file system could allocate 4 KB of disk space as a single 4-KB block or as eight 512-byte blocks. How could we take advantage of this flexibility to improve performance? What modifications would have to be made to the free-space management scheme in order to support this feature?

3. Fragmentation on a storage device could be eliminated by recompaction of the information. Typical disk devices do not have relocation or base registers (such as are used when memory is to be compacted), so how can we relocate files? Give two reasons why recompacting and relocation of files often are avoided.

4. Discuss how performance optimizations for file systems might result in difficulties in maintaining the consistency of the systems in the event of computer crashes.

1.

①随机访问更快：不必读磁盘就能知道下一个块是谁

②数据块不花费空间存储指针：每块全是数据

③查找空闲块快

④顺序读文件更快：不用物理跳转去读取链指针

2.

性能优化：①适配文件大小：小文件分配小粒度文件，避免大粒度文件造成的内部碎片，大文件分配大粒度块(4KB)，减少块的总数量，降低开销。

②平衡访问与空间利用率：根据文件的预期大小和访问模式动态选择粒度。

修改：

使用分级空闲链表：为不同的块分别维护空闲链表，分配时根据文件需求选择对应粒度的空闲块。

3.

方法：通过修改文件系统的元数据实现：例如更新文件的索引

块、文件分配表中的块映射关系。无需硬件寄存器支持，而是通过文件系统自身的结构记录并更新块的位置

原因：① 性能开销大：重整理需要大量I/O操作，严重占用带宽，导致性能下降

② 并发与可靠性风险：若文件被其他进程打开或访问，重定位过程中需处理复杂的并发同步问题。可能崩溃，导致数据丢失。

4.

当采用延迟更新来优化系统性能，如果系统在没有提交延迟更新的情况下崩溃，那么文件系统的一致性就会被破坏。

性能优化意味：① 延迟写回

② 批量更新元数据

③ 日志异步写入

④ 元数据不立即同步

实际磁盘状态并不总是与内存状态一致，缓存写导致崩溃时数据不同步。

可能导致：① 目录项已写入，但数据块没写入 ——→ 文件损坏

② 多个元数据更新不同步→文件系统不一致。