作业六

> 22373407　王飞阳

1.

- 先来先服务，磁头移动的次序为：

  $15 \rightarrow 10 \rightarrow 35 \rightarrow 20 \rightarrow 70 \rightarrow 2 \rightarrow 3 \rightarrow 38$

  寻道总时间：(5+25+15+50+68+1+35) * 5 = 995ms

- 最短寻道时间优先，磁头移动的次序为：

  $15 \rightarrow 10 \rightarrow 3 \rightarrow 2 \rightarrow 20 \rightarrow 35 \rightarrow 38 \rightarrow 70$

  寻道总时间：(5+7+1+18+15+3+32) * 5 = 405ms

- 查看SCAN算法算法，磁头移动的次序为：

  $15 \rightarrow 20 \rightarrow 35 \rightarrow 38 \rightarrow 70 \rightarrow 85 \rightarrow 10 \rightarrow 3 \rightarrow 2$

  寻道总时间为：(5+15+3+32+60+7+1)*5 = 765ms

- 查看扫描Look算法，磁头移动的次序为：

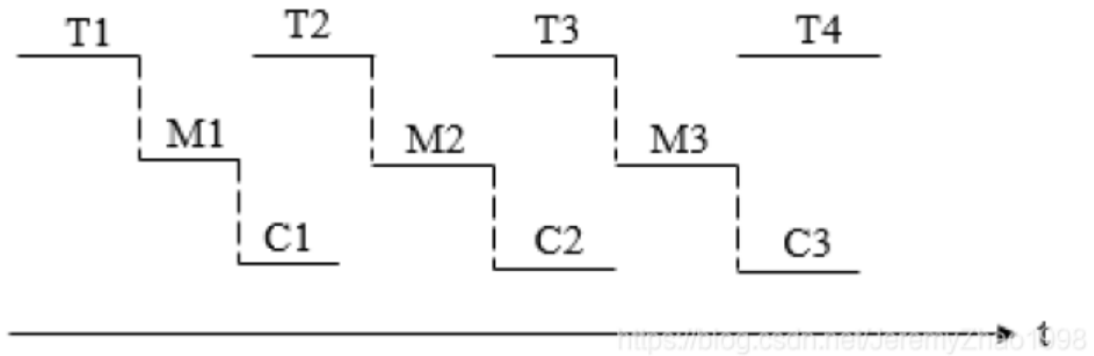
  $15 \rightarrow 20 \rightarrow 35 \rightarrow 38 \rightarrow 70 \rightarrow 10 \rightarrow 3 \rightarrow 2$

  寻道总时间为：(5+15+3+32+60+7+1)*5 = 615ms

2.

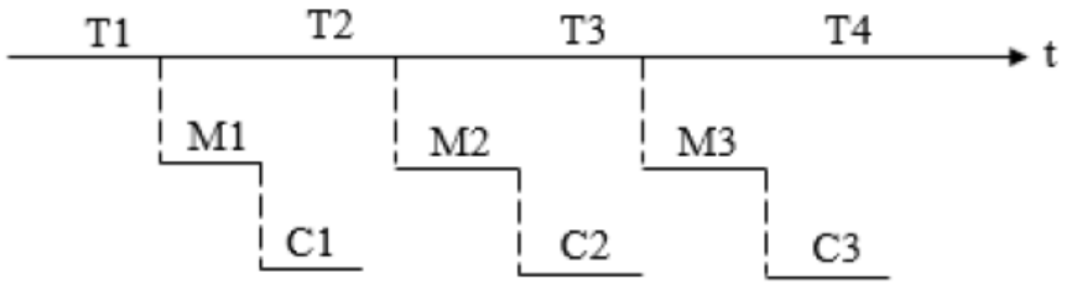
引入缓冲技术的原因：提高外设利用率，匹配CPU与外设的不同处理速度，减少对CPU的中断次数，提高CPU和I/O设备之间的并行性。

单缓冲结构下数据IO的过程示意如下，其中T代表读入缓冲区的时间，M代表缓冲区传送到用户区的时间，C代表CPU对数据分析的时间：



每一个磁盘块数据需要的时间为$max(T,C)+M$，读入时间T大于处理时间C，因此每一个磁盘块数据需要的时间为T+M=150μs，所有数据完成的时间为(T+M)*10+C=1550μs

双缓冲结构下数据IO的过程示意如下，其中T代表读入缓冲区的时间，M代表缓冲区传送到用户区的时间，C代表CPU对数据分析的时间：



在CPU传输和处理一个缓冲区数据的过程中，IO设备可以向另一个缓冲区读入数据。由于M和C的时间之和恰好等于T，每一个数据块需要的时间只有T=100μs，所有数据完成的时间为T*10+M+C=1100μs

3.

1. **磁盘I/O优化**：
   - **预读取和延迟写入**：文件系统可以实施预读取（read-ahead）和延迟写入（write-back）策略。预读取是指在读取某个数据块时，系统预测后续可能会读取的数据块，并提前将它们载入内存。延迟写入则是将修改过的数据暂存于内存中，等到最优的时机再一次性写回磁盘，减少磁盘操作次数。
   - **磁盘排列**：根据文件访问的局部性原理，可以将经常一起访问的文件或数据块在磁盘上相邻地放置，以减少磁盘寻道时间。
2. **缓存管理**：
   - **智能缓存算法**：采用更高效的缓存替换算法（如LRU、ARC等），以保持最常访问的数据在缓存中。这可以显著减少对磁盘的访问需求。
   - **缓存分层**：实现多级缓存机制，例如，在内存中维护一个主缓存，在SSD等快速存储设备上维护次级缓存，进一步提高数据访问速度。
3. **数据结构的选择**：
   - **B树及其变种**：文件系统中广泛使用B树（如B+树、B*树）等数据结构来组织和索引文件数据。这些树状结构优化了节点的分裂和合并过程，有助于保持高效的查找和更新速度。
   - **散列表**：对于需要快速查找的元数据，如inode映射，使用散列表可以提高访问效率。
4. **并发控制**：
   - **细粒度锁**：实施更细粒度的锁策略（如基于inode的锁），以允许更多的并发操作，减少锁竞争。
   - **锁的层次化**：设计一个层次化的锁系统，不同级别的操作使用不同级别的锁（如全局锁、文件锁、块锁等），以提高并发性能。
5. **日志和事务管理**：
   - **日志结构文件系统**：通过维护一个操作日志来记录文件系统的修改，可以在系统崩溃后快速恢复。日志先写策略也可以优化写操作的性能。
   - **元数据事务**：对元数据操作使用事务机制，确保文件系统的一致性和可靠性。
6. **智能错误处理**：
   - **冗余和校验**：采用RAID技术或文件系统内置的校验和修复机制（如ZFS中的校验和自动修复功能），可以提高数据的可靠性和错误恢复能力。

4.

- 基本信息
  - 文件名：字符串，通常在不同系统中允许不同的最大长度，可修改
  - 物理位置
  - 文件逻辑结构：有/无结构（记录文件，流式文件）
  - 文件物理结构：（如顺序，索引等）
- 访问控制信息
  - 文件所有者（属主）：通常是创建文件的用户，或者改变已有文件的属主
  - 访问权限（控制各用户可使用的访问方式）：如读、写、执行、删除等
- 使用信息
  - 创建时间，上一次修改时间，当前使用信息等。

5.

1. 访问二级目录：根目录的目录项已经读入内存，那么读取二级目录不需要访问磁盘。

   访问三级目录：一个磁盘块1KB，每个目录项128B，那么一个磁盘块可以放1KB/128B=8个目录项；而如图所示每个二级目录下有128个三级目录，这些三级目录分布在128/8=16个磁盘块上。串联文件形式，访问一个三级目录项至少访问1次磁盘，至多访问16次磁盘，平均8.5次。

   取出页目录项：第三级目录的目录项可以一次从磁盘读出，因此1次从磁盘中取出对应得页目录项。

   访问文件块：文件平均大小100KB，每个磁盘块1KB，每个文件平均要分布在100KB/1KB=100个磁盘块上。串联文件形式，访问一个块至少访问1次磁盘，至多访问100次，平均50.5次。

   综上，平均共需要访问磁盘1+8.5+50.5=60次。

2. 一个目录项只占14+2=16个字节，那么一个磁盘块可以存放1KB/16B=64个目录项。

   读取根目录：由于根目录inode已在内存中，根目录下只有3个页目录项，可以1次读取。

   读取二级目录：读取user2的inode需要读取1次磁盘，读取user2目录的内容，最多需要访问磁盘128/64=2次，平均（1+2）/2 = 1.5次。

   读取三级目录：读取三级目录的inode需读取1次磁盘，读取三级目录的文件，由于三级目录的文件不超过50个，因此可以1次读出。

   读取文件：读取文件的inode需要读取1次磁盘，读取文件由于采用直接索引，故可根据inode可直接读取磁盘上文件的一个块，需要读取1次。

   综上，平均共需要访问磁盘1+1+1.5+1+1+1+1=7.5次。

3. 该文件系统管理的数据块数为$16ZB/1KB = 2^{64}$块。表示这些磁盘块需要64/8=8字节，故索引区可存放512/8 = 64个磁盘块号。一级索引指向的磁盘块中可存储1KB/8B=128个磁盘块号，在采用一级索引的情况下，支持的最大文件为$64 * 128 * 1KB = 8MB$。