操作系统作业 5

1.

a) RAID-5 更新一个数据块需要访问5个块（读3个块，写2个块）；RAID-6更新一个数据块需要访问5个数据块（读2个块，写3个块）。

b) RAID-5 更新7个连续数据块需要访问10个块（读1个块，写7+2个块）；RAID-6更新连续7个数据块需要访问15个数据块（读2个块，写7+2*3个块）

2.

- a. FCFS(先到先服务)：`2150、2069、1212、2296、2800、544、1618、356、1523、4965、3681`

总距离为：abs(2150-2069)+abs(2069-1212)+abs(1212-2296)+abs(2296-2800)+abs(2800-544)+abs(544-1618)+abs(1618-356)+abs(356-1523)+abs(1523-4965)+abs(4956-3681)=**13002**

- b. SSTF（最短寻道时间优先）：`2150、2069、2296、2800、3681、4965、1618、1523、1212、544、356`

总距离为：abs(2150-2069)+abs(2069-2296)+abs(2800-2296)+abs(3681-2800)+abs(3681-4965)+abs(4965-1618)+abs(1618-1523)+abs(1212-1523)+abs(1212-544)+abs(544-356)=**7586**

- c. SCAN(扫描算法，磁头方向反转)：前一个是1805<2150 说明磁头方向朝着柱面大的方向移动 `2150、2296、2800、3681、4695、(4999)、2069、1618、1523、1212、544、356`

总距离为：abs(2150-2296)+abs(2296-2800)+abs(2800-3681)+abs(3681-4965)+abs(4965-9999)+abs(9999-2096)+abs(2096-1618)+abs(1618-1523)+abs(1212-1523)+abs(1212-544)+abs(544-356)=**7492**

- d. LOOK(磁头反转，和SCAN的区别是不需要移动到首尾就可以反转，到达等待列表中的最大值或最小值柱面就可以反转)：`2150、2296、2800、3681、4695、(空，和SCAN的区别)、2069、1618、1523、1212、544、356`

总距离为：abs(2150-2296)+abs(2296-2800)+abs(2800-3681)+abs(3681-4965)+abs(4965-2086)+abs(2096-1618)+abs(1618-1523)+abs(1212-1523)+abs(1212-544)+abs(544-356)=**7424**

- e.C-SCAN(磁头反转跳到另一端)：`2150、2296、2800、3681、4695、(4999)、(0)、356、544、1212、1523、1618、2069`

总距离为：abs(2150-2296)+abs(2296-2800)+abs(2800-3681)+abs(3681-4965)+abs(4965-4999)+abs(4999-0)+abs(0-356)+abs(356-544)+abs(544-1212)+abs(1212-1523)+abs(1523-1618)+abs(1618-2069)=**9917**

- f. C-LOOK(磁头反转跳到另一端，左右反转断点不一定是0和499)：`2150、2296、2800、3681、4695、356、544、 1212、1523、1618、2069`

总距离为：abs(2150-2296)+abs(2296-2800)+abs(2800-3681)+abs(3681-4965)+abs(4965-356)+abs(356-544)+abs(544-1212)+abs(1212-1523)+abs(1523-1618)+abs(1618-2069)**9137**

3.

操作系统使用系统调用open()打开一个文件，与这个打开文件相关的各种信息（包括文件偏移量，状态标志信息，i-node 对象指针）都有一个系统级别的文件打开表来存储和维护，这个表就是打开文件表。

使用打开文件表可以节省开销，提高性能。因为已经打开的文件的信息位于打开文件表中，而在打开文件表完成文件索引所需的开销远远小于遍历目录进行索引。并且在全局层面上维护一个打开文件表可以简化文件删除等操作的实现难度。加入没有打开文件表，两个进程可能对同一个文件同时进行写入和删除，这是危险的，打开文件表使得避免这种情况变得简单。

4.

755：表示文件权限

7(10)=111(2),表示用户（文件拥有者）权限，允许读、写、执行

5(10)=101(2),表示文件所属组的其它用户的权限，允许读和执行，但不能写（修改）

5(10)=101(2),表示组外的其它用户的权限，允许读和执行，不能写

5.

连续分配会产生外部碎片的问题。随着文件的分配和删除，可用磁盘空间被分成许多小片。只要空闲空间分成小片就会存在外部碎片。当最大的连续片不能满足需求时就有问题。

外部碎片的解决方法是：

- 将所有空闲空间进行 **合并**，但这种合并的代价是时间。
- 当然另一种解决方法就是不使用连续分配，而是将文件分块存储。

连续分配的另一个问题是，确定一个文件需要多少空间。用户创建一个文件时，往往不知道这个文件可能会有多大，如果文件分配空间太少，则可能导致文件无法扩展。

文件拓展问题的解决方法是：

- 终止用户程序，给出适当错误信息。用户必须分配更多的空间。这种方法理论可信，但相信这种操作系统不会受消费者满意。
- 另一种方式是，找一个更大的空间，将文件内容复制到新空间，并释放原来的空间

6.

使用FAT的链接分配的优点是改善了随机访问的时间。因为通过读入FAT信息，磁头能找到任何块的位置。这使得访问文件中间部分的块时，可以查找存储在 FAT 中的指针来确定其位置，而非按顺序访问文件的所有块来找到指针指向目标块的指针

FAT的主要问题是：需要对FAT使用缓存，FAT存储在内存中，使用空间换取时间效率，内存空间的浪费就难以避免。

7.

a. root direct - - >inode /a - - > block /a - - > inode /a/b - - > block /a/b - - > inode /a/b/c - - > block /a/b/c 需要从7次磁盘I/O

b. root direct - - >inode /a - - > block /a - - > inode /a/b - - > block /a/b - - > inode /a/b/c - - > block /a/b/c 其中inode 都在缓存中，所以只需要7-3=4次磁盘I/O

8.

指针4字节，则地址有10位，2^{10}=1024

8kb*(12+1024+1024^2+1024^3)=8598331488 kb

即文件最大位**8598331488** kb

9.

Hard Links： 为已存在的文件，新建一个指向该文件inode节点的目录项（directory entry），同时增加该inode的连接数。可以让一个文件有多个硬链接，出现在多个目录下。实际上是一个文件有多个路径。

Symbolic Links：跟hard link不同，**它是建立一个独立的文件，而这个文件的作用是当读取这个连接文件时，它会把读取的行为转发到该文件所link的文件上。**symbolic links 创造了一个新的目录条目和inode,它的目录条目指向的是这个新的inode,这个inode里存储了指向文件的路径。例如，现在有文件a，我们做了一个软连接文件b（只是一个连接文件，非常小），b指向了文件a。当读取b时，那么b就会把读取的动作转发到a上，这样就读取到了文件a。所以，删除文件a时，文件b并不会被删除，但是再读取b时，会提示无法打开文件。而当你删除b时，a是不会有任何影响的。

10.

数据日志和元数据日志的区别：

数据日志会将元数据和数据都写入journal，而元数据日志不写入数据，而只写元数据。数据日志需要将数据写入磁盘两次，一次是commit to journal file,另一次是checkpoint。而元数据日志取消了第一次写入。

数据日志的操作顺序：

- journal write : 将TxB,metadata,data写入journal
- journal commit:提交metadata和data（包括TxE)
- checkpoint:将更新的内容写入它们在磁盘上的位置
- free:清理空间（清楚journal中的metadata和data）

元数据日志的操作顺序：

- data write
- journal metadata write(可与前一个并行执行)
- journal commit
- checkpoint metadata
- free

11.

三种I/O控制方式：

- 轮询（Polling)
- 中断（interrupt）
- 直接内存访问（DMA)

12.

1.I/O 调度(I/O scheduling)

2.缓冲(Buffering)

3.缓存(Caching)

4.假脱机与设备预留(Spooling)

5.错误处理（Error handling）

6.I/O 保护(I/O protection)