操作系统作业 5

PB19051183 吴承泽

1.

a.

RAID-5 需要访问 2 个块

RAID-6 需要访问 3 个块

b.

RAID-5 需要访问 9 个块

RAID-6 需要访问 13 个块

2.

a.FCFS

total time t = 81 + 264 + 857 + 1084 + 504 + 2256 + 1074 + 1262 + 1167 + 3442 +1284 = 13275

b.SSTF

total time t = 54 + 227 + 504 + 881 + 1284 + 3347 + 95 + 311 + 668 + 188 = 7559

c.SCAN

total time t = (4999 – 2150) + (4999 - 356) = 7492

d.LOOK

total time t = (4965 - 2150) + (4965 - 356) = 7424

e.C-SCAN

total time t = (4999 - 2150) + 4999 + 2069 = 9917

f.C-LOOK

total time t = (4965 - 2150) + (4999 - 356) + (2069 - 356) = 9197

3.

打开文件表(open-file table)是操作系统用于维护所有打开文件信息的一个查找表。

打开文件表中存储了每个进程打开文件的文件信息，其中通过打开文件表可以快速检索进程所需要的文件信息，在读写时不需要进行目录的遍历。

4.

文件：对 OS 系统来说，文件是一种逻辑上的存储单元，是一种抽象的数据概念；对用户空间来说，是一个最小的逻辑存储分配。

目录：也是一种文件，只不过内部存放的是子目录与目录中的文件名。

“755”表示对所有者有可读、可写、可执行三类权限，对用户组和其他用户拥有可读可执行的权限。

5.

对文件系统来说，使用连续的磁盘分配方式会使一些占用存储空间大文件无法在磁盘中找到足够大的连续地址，导致无法加载到磁盘中，即使仍有足够的空间；而且对于文件来说，连续分配有可能会导致该文件无法在文件系统中增长；同时由于连续分配，会产生磁盘碎片等，造成资源浪费。

解决方式：将磁盘分区，分为一个个块，通过 FAT 表记录对应文件的块以及其对应的下一个块来查找每个文件的数据存储位置。

6.

优点：性能好，检索速度较快，且生成文件，文件成长，删除文件时性能都十分优秀

主要问题：整个 FAT 表都需要存储在内存中，会浪费一部分内存空间

7.

Workflow：首先需要访问 Root Directory,查找到所需的根目录文件夹 a 后，访问 Index Node Table,查找到地址后找到 Data Block 中的 a 文件夹,通过该 Data Block 存储的 b 文件夹 INode 编号访问 Index Node Table，找到 b 文件夹对应的 Index Node，找到对应的 b 文件夹的 Data Block 域，再在 b 文件夹找其中的 c 文件，在其 Directory file 中搜索 c 文件，并通过 INode 编号访问 Index Node Table，通过地址访问 c 文件的 Data Blocks，并完成读操作。

a.假定没有块被放入缓存中时，则每次访问 Index Node Table 都需要进行 IO 操作，则一共需要 7 次

b.假定 INode 都在内存中，则访问 Index Node Table 时不需要做 IO 操作，则一共需要 4 次

8.

$$(12 + 2^{13-2} + 2^{2*(13-2)} + 2^{3*(13-2)}) * 8\,KB \approx 64TB$$

9.

8+3 即文件名占八个字节，扩展名占三个字节，若是遇到长文件名，则在目录下增加 LFN Entry 的结构体来存储额外的文件名。

10.

Directory Entry 是一种数据结构，其中包含了为寻找文件数据块的索引与该文件的信息，其中前 11 字节存储文件名与扩展名，方便查找，其中在固定的地址上存储了数据块的簇号，以及该文件的大小。若是 IO 需读一个文件，可以通过查找该文件的 Dir Entry，定位数据块的首个簇号和文件大小来进行该文件的数据访问。

11.

Hard Link 相当于产生一个新的路径，不会创建新的文件内容，也不会创建新的 Inode；Symbolic Link 相当于产生了一个新的文件，会创建一个新的 Inode 并指向链接的数据域。

12.

一个文件被创建时 link counts 的数目为 1，一个目录被创建时 link counts 的数目为 2。

正常创建文件时，在其目录下会产生一个 Dir Entry 分配一个新 Inode，直接指向该文件，这样子指向 Inode 的连接仅有文件所在目录的那一个；而正常创建目录时，除了父目录中的一个连接，新创建的目录中会有文件名为'.'的路径指向自己目录的 Inode，从而有两个连接指向新创建文件的 Inode。

13.

Data Journaling 与 Metadata Journaling 的区别在于 Data Journaling 需要将数据块写入日志中，而 Metadata Journaling 不需要将 Db 写入日志里。

Data Journaling 的操作顺序为，先写入 Txb，Inode，bitmap 和 Db（数据与元数据）至日志中，再将 TxE 写入日志中，再将 metadata 和 data 更新至文件系统的具体在的位置，最后将日志该空间释放或者等待日志覆盖这个位置。

Metadat Journaling 的操作顺序为，并行写入新数据和日志中的元数据以及 TxB，再在日志中写入 TxE，再写入 metadata 至文件系统的应该所在的位置，最后释放空间或等待日志覆盖。

14.

轮询、中断、直接内存访问

15.

存储设备、传输设备和人机交互设备。

包括 硬盘，鼠标或互联网等。

方法：设计抽象、封装和软件分层。内核可以从 I/O 设备抽象出一些通用类型，通过一组标准接口来访问，各种各样的 I/O 设备的差异可以被封装到定制的设备驱动中以适应不同的设备，同时提供标准接口对接内核或内核子系统。

16.

I/O 调度，提供缓冲区，缓存，假脱机与设备预留，错误处理，I/O 保护以及电源管理