

操作系统作业 7

PB18151866 龚小航

1. What is the 8+3 naming convention in FAT32 file system, and how to manage long filenames?

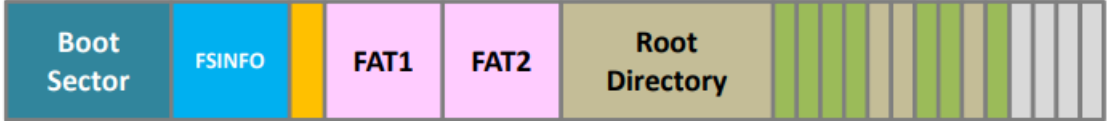
解：在 FAT32 文件系统中，8+3 的短文件命名规则指：文件名总是八个字（Byte）不足则填充空格；文件扩展名总是 3 个字，不足填充格。若第一位为 0x00 或 0xE5，则表示未分配。

- 当文件名比较长的时候，条目就需要使用 LFN（Long File Name）。区分普通条目和 LFN 条目的标识符在于第 11 个字节，LFN 条目一般将其设为 0x000F。一个 LFN 条目可以包含 13 个 unicode 形式的字符（即 26 Byte）。若一个长文件名不足以保存所有的文件名以及其扩展名，则继续分配下一个长文件名，直到可以完全容纳，LFN 形式条目的第一个字节就是用于存放条目顺序的。需要注意的是，若使用了多个长文件名条目，则它在目录文件中是倒序存放的。
- 两种条目的具体结构分布如下所示：

Normal entry		LFN entry	
Bytes	Description	Bytes	Description
0-0	1 st character of the filename (0x00 or 0xe5 means unallocated)	0-0	Sequence Number
1-10	7+3 characters of filename + extension.	1-10	File name characters (5 characters in Unicode)
11-11	File attributes (e.g., read only, hidden)	11-11	File attributes - always 0x0F
12-12	Reserved.	12-12	Reserved.
13-19	Creation and access time information.	13-13	Checksum
20-21	High 2 bytes of the first cluster address (0 for FAT16 and FAT12).	14-25	File name characters (6 characters in Unicode)
22-25	Written time information.	26-27	Reserved
26-27	Low 2 bytes of first cluster address.	28-31	File name characters (2 characters in Unicode)
28-31	File size.		

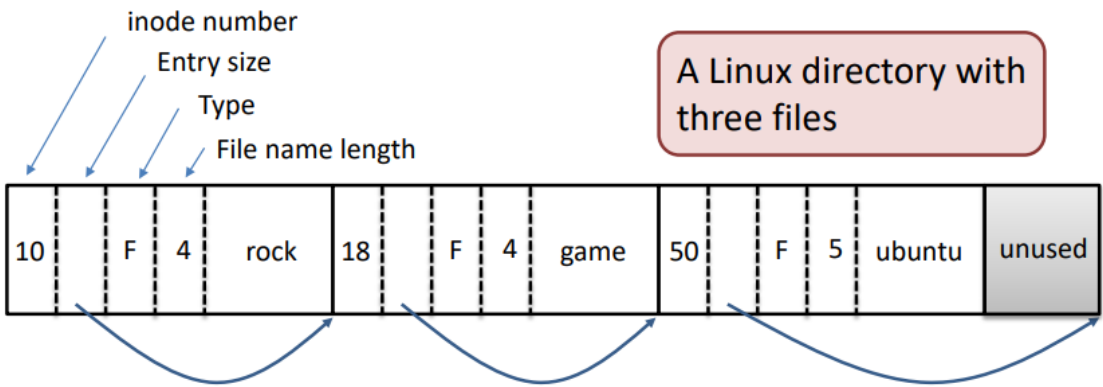
2. How are director entries managed in FAT and Ext file systems?

解：在 FAT32 文件系统下，每个目录条目都是一个 32Byte 的结构（如上图所示）。这些目录条目存放在各级目录下。目录条目存储了文件名和文件属性。



当需要访问一个文件时，先从 Root Directory 开始，根据路径在某个目录下找到需要的条目，在条目中可以获取这个文件的起始块位置；再到 FAT 表中找对应的下一块，直到找到结束标识为止。

在 EXT 文件系统下，目录条目存放了文件名以及 inode，文件名信息和属性地址在目录条目里面，属性的内容单独放在 inode 里面，目录的结构如下所示：



FAT 里目录的结构相当于数组结构，EXT 里目录结构相当于链表结构

3. What is the difference between hard link and symbolic link?

解：硬链接和符号链接的区别：

- 硬链接：硬链接是指向已有文件的一个目录条目。在原有的目录条目之上，再加一个新的条目，它的路径不同而指向的是相同的文件。即条目中 inode 的值是一样的，没有创建新的文件。
- 符号链接：当创建一个新的符号链接时，事实上创建了一个新的 inode，而这个新的 inode 里存的就是被链接文件的路径。一个 inode 块中能存放 60 个字符的路径长度。

4. What are the initial link counts when a regular file or a directory is created? Why?

解：

- 当创建一个普通文件的时候，即为在这个文件所在的目录下面创建条目，若不对其作人为的链接，此时的链接数就是 1；
- 当创建一个目录文件时，初始的链接数为 2：创建目录时除了与常规文件一样的在其父目录中有一个链接，目录文件还固有两个子条目“.”“..”其中前者硬链接至这个目录文件本身，而后者硬链接至其父目录。因此创建一个目录文件时，初始的链接数为 2。

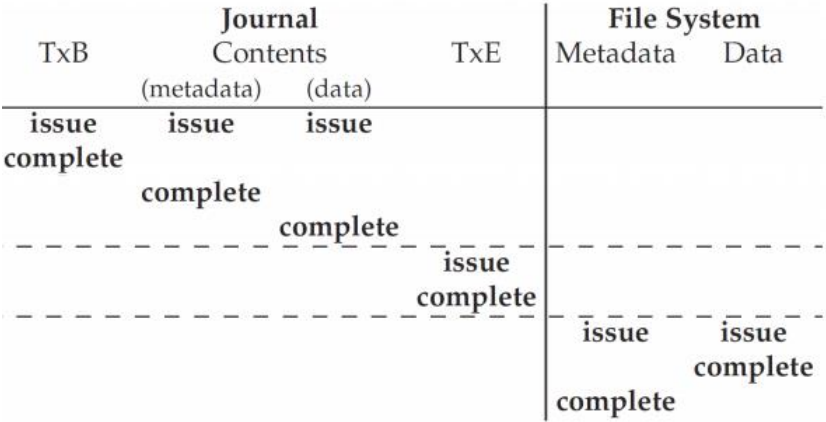
5. What is the difference between data journaling and metadata journaling? Explain the operation sequence for each of the two journaling methods.

解：数据日志和元数据日志的区别：

Data 数据指用户需要的数据，而 Metadata 指 inode, bitmap 这一类数据。

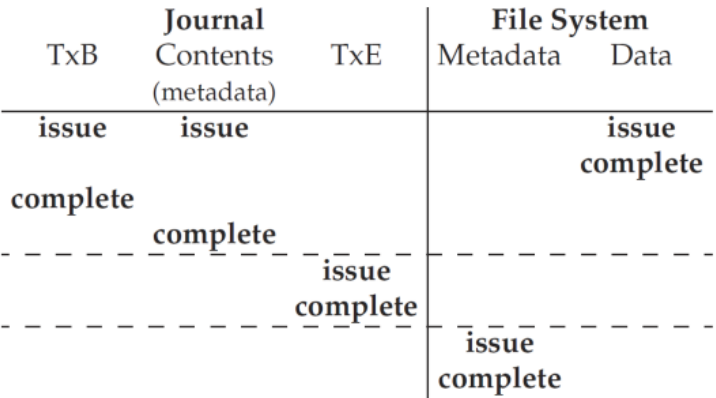
- 数据日志：特点是写入日志的是所有的数据，包括 Data 和 Metadata。首先先做写日志操作 (Journal Write)，写完成之后做日志提交 (Journal Commit)，提交完成后进行检查点，即将日志中记录的更改写进实际的文件系统当中去。

数据日志的操作顺序如下图所示：虚线规定了两侧的执行顺序（强制同步）



- 元数据日志：由于数据日志需要将数据和元数据一起记入日志中，那么 data block 就被写了两次，这对性能是一个比较大的影响。在一般系统中，数据量比元数据的量大很多。元数据日志就是一种改进，在记录日志文件时不将 data block 块写入，而仅仅将元数据写入日志。元数据日志在提交之前，必须先将目标数据块更改完成，否则若先提交，在写回数据的过程中系统崩溃则会得到一块垃圾数据。

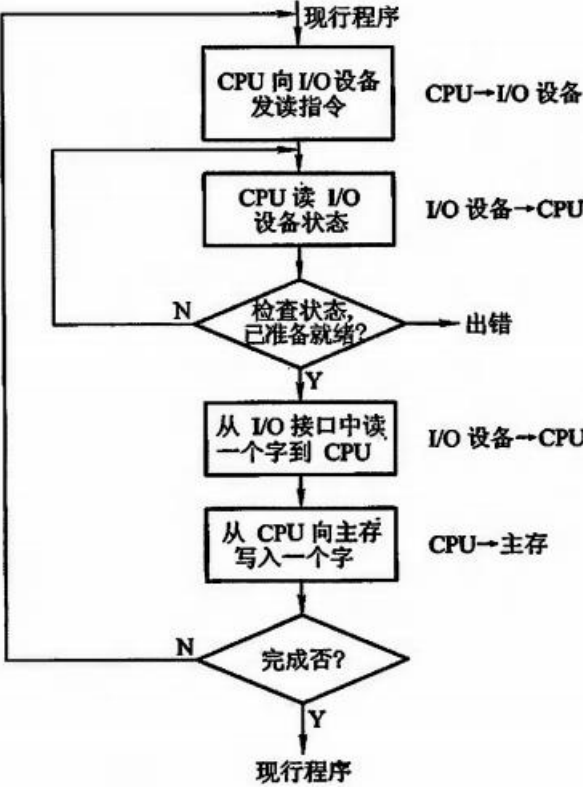
元数据日志的操作顺序如下图所示：虚线规定了两侧的执行顺序（强制同步）



6. What are the three I/O control methods?

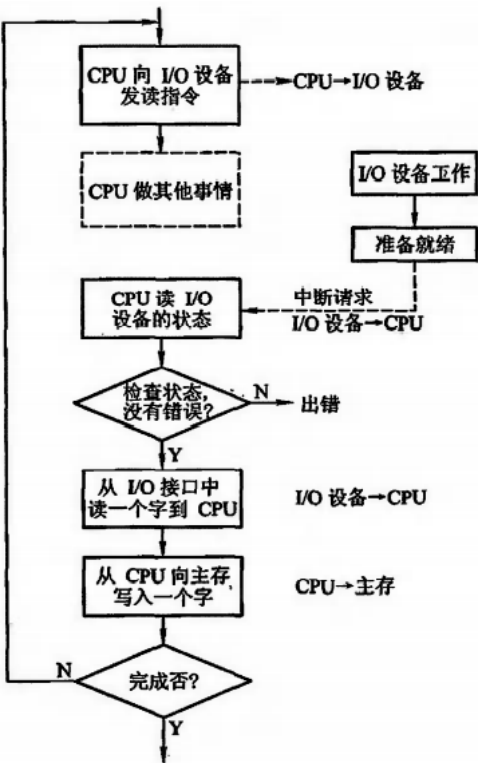
解：三种 I/O 控制方式分别为 轮询、中断、直接内存访问

- 轮询：程序查询方式是由 CPU 通过程序不断查询 I/O 设备是否已做好准备，从而控制 I/O 设备与主机交换信息。采用这种方式交换信息要求 I/O 接口内设置一个准备是否就绪的状态标记。程序查询方式的流程如左图所示。CPU 启动 I/O 设备后便开始对 I/O 设备的状态进行查询，若未就绪则会继续查询。



可见，只要一启动 I/O 设备，CPU 就不断查询 I/O 设备的准备情况，从而终止原程序的执行；另一方面，CPU 要一个字一个字的从 I/O 设备取出数据，经 CPU 送至主存，这个过程中 CPU 也不能执行原程序。程序查询方式使 CPU 的操作和外围设备的操作能够同步，但浪费了很多 CPU 时间。这种方式使 CPU 和 I/O 设备处于串行工作状态，CPU 的工作效率不高。

- 中断：中断方式是轮询方式的一个改进。CPU 在启动 I/O 设备后，不查询设备是否已准备就绪，就继续执行原程序，当且仅当 I/O 设备准备就绪并向 CPU 发出中断请求后才予以响应。



左图是采用中断方式交换数据的流程图。在这种方式中，I/O 设备通过中断告知 CPU 外设已经准备就绪，而 CPU 不必一直查询 I/O 设备的准备情况，而是继续执行原程序，没有在这上面浪费 CPU 时间；中断到来时，CPU 处理中断，将一个字从 I/O 中写入主存，然后继续执行原程序；若是有一批数据，则需要多次发起中断，每次 CPU 向主存写入一个字。相比程序查询方式，采用程序中断方式相比于采用程序查询方式大大提高了 CPU 的工作效率。

图 5.11 程序中断方式流程

- 直接内存访问 (DMA 方式)：DMA 方式是对程序中断方式的又一次提高。程序中断方式虽然不需要反复查询 I/O 设备是否就绪，但 CPU 仍需要响应中断请求，停止现行程序并执行中断服务程序，为了交换信息还不得不占用一些 CPU 内的寄存器，这些同样是对 CPU 资源的消耗。DMA 方式提出 I/O 设备直接与主存交换信息而不占用 CPU，这种方式称为直接存储器存取方式。

在 DMA 方式中，I/O 设备直接与主存连有数据通路，交换信息时无需调用中断服务程序。若 CPU 和 DMA 同时访问主存，CPU 总会将总线占有权让给 DMA。在这个窃取周期内，CPU 依然可以做一些内部操作。可见，与程序查询和程序中断方式相比，DMA 方式进一步提高了 CPU 的资源利用率。

下图总结了三种数据交换方式的 CPU 工作效率比较：

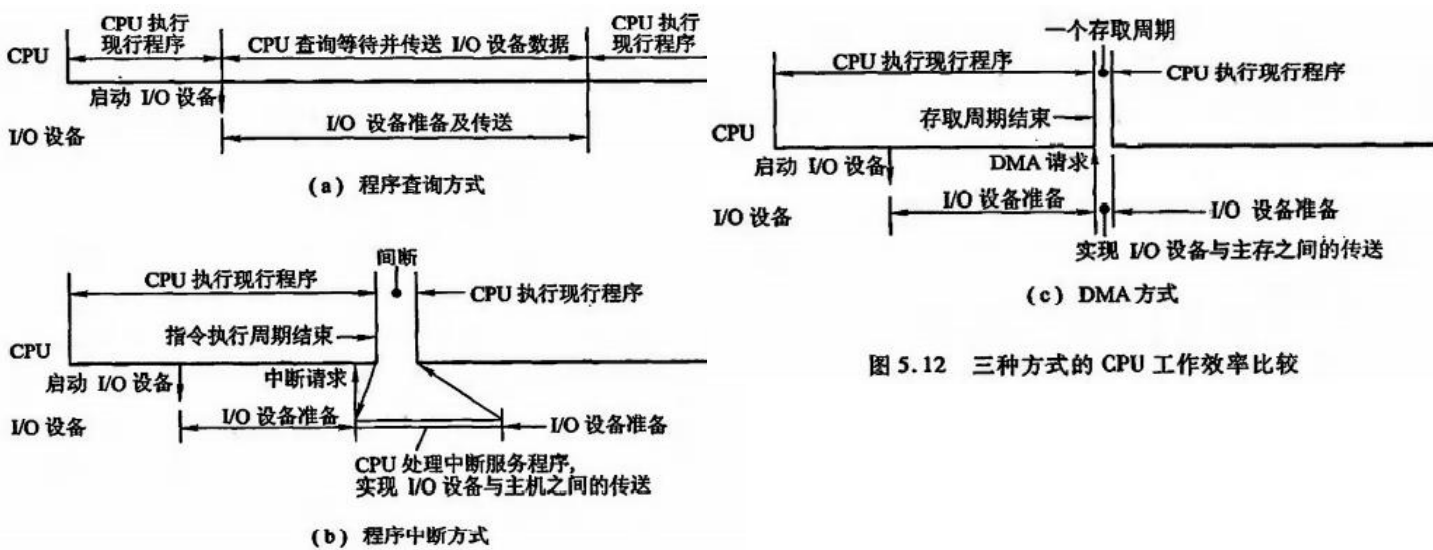


图 5.12 三种方式的 CPU 工作效率比较

7. List at least three kinds of I/O devices and explain how to provide a standard and uniform application I/O interface?

解：列出一些 I/O 设备：硬盘，显示器，键盘，鼠标等。

- 为了封装各种设备的细节与特点，操作系统内核采用设备驱动程序模块。**设备驱动程序**为 I/O 子系统提供了统一的设备访问接口。

8. What services are provided by the kernel I/O subsystem?

解：内核提供许多与 I/O 相关的服务：**调度、缓冲、缓存、假脱机、设备预留及错误处理**。这些服务还建立在硬件和设备驱动程序的基础设施之上。I/O 子系统也负责保护自己免受错误进程和恶意用户的侵扰。

- I/O 调度：调度一组 I/O 请求即是确定一个好的顺序来执行它们。一个好的调度可以减少 I/O 完成所需的平均等待时间。I/O 调度服务是 I/O 子系统提高计算机效率的一种方法。
- 缓冲：缓冲区是内存的一块区域，用于保存在两个设备之间或在设备和应用程序之间传输的数据。缓冲服务有三大作用：
 - ① 能够处理数据流的生产者与消费者之间的速度不匹配；
 - ② 能够协调传输大小不一数据的设备；
 - ③ 支持应用程序 I/O 的复制语义。
- 缓存：缓存是保存数据副本的高速内存区域。访问缓存副本比访问原版更加有效。
- 假脱机与设备预留：假脱机是保存设备输出的缓冲区，某些设备（如打印机）不能接收交叉的数据流。
- 错误处理：作为一般规则，I/O 系统调用通常返回一位的调用状态信息，以表示成功与否。

I/O 子系统还提供一些其他的 service，例如 I/O 保护以及电源管理等。