

第 2 章 文件管理习题解答

1. 什么是文件和文件系统？文件系统有哪些功能？

【解答】文件是具有符号名而且在逻辑上具有完整意义的信息项的有序序列。

文件系统是指操作系统系统中实现对文件的组织、管理和存取的一组系统程序，它实现对文件的共享和保护，方便用户“按名存取”。

文件系统的功能“

(1) 文件及目录的管理。如打开、关闭、读、写等。

(2) 提供有关文件自身的服务。如文件共享机制、文件的安全性等。

(3) 文件存储空间的管理。如分配和释放。主要针对可改写的外存如磁盘。(4) 提供用户接口。为方便用户使用文件系统所提供的服务，称为接口。文件系统通常向用户提供两种类型的接口：命令接口和程序接口。不同的操作系统提供不同类型的接口，不同的应用程序往往使用不同的接口。

2. Linux 文件可以根据什么分类？可以分为哪几类？各有什么特点？

【解答】在 Linux 操作系统中，文件可以根据内部结构和处理方式进行分类。

在 Linux 操作系统中，可以将文件分为普通文件、目录文件、特别文件三类。

各类文件的特点是：

普通文件：由表示程序、数据或正文的字符串构成的文件，内部没有固定的结构。这种文件既可以是系统文件，也可以是库文件或用户文件。

目录文件：由文件目录构成的一类文件。对它的处理(读、写、执行)在形式上与普通文件相同。

特别文件：特指各种外部设备，为了便于管理，把所有的输入/输出设备都按文件格式供用户使用。这类文件对于查找目录、存取权限验证等的处理与普通文件相似，而其他部分的处理要针对设备特性要求做相应的特殊处理。

应该指出，按不同的分类方式就有不同的文件系统。

3. 什么是文件的逻辑结构？什么是文件的物理结构？Linux 文件系统分别采用什么样的结构？有什么优点和缺点？

【解答】文件的逻辑结构：用户对文件的观察的使用是从自身处理文件中数据时采用的组织方式来看待文件组织形式。这种从用户观点出发所见到的文件组织方式称为文件的逻辑组织。

文件的物理结构：从系统的角度考察文件在实际存储设备上的存放形式，又称为文件的存储结构。

在 Linux 系统中，所有文件的逻辑结构都被看作是流式文件，系统不对文件进行格式处理。

在 Linux 系统中，文件的物理结构采用的是混合多重索引结构，即将文件所占用盘块的盘块号，直接或间接地存放在该文件索引结点的地址项中。

在 Linux 系统中，采用混合索引结构的优点是，对于小文件，访问速度快；对于大中

型的文件，其文件系统也能很好的支持；缺点是会带来因为多次访问磁盘而影响速度。

4. 目前广泛用的目录结构形式是哪种？它有什么优点？

【解答】广泛用的目录结构形式是树型目录结构。

优点：

- (1) 能有效的提高对目录的检索速度。
- (2) 允许文件重名。允许用户在自己的分目录中，使用与其他用户相同的文件名。
- (3) 便于实现文件共享。

5. 常见的文件存储空间管理有哪几种？各自有什么特点？Linux 系统的存储空间管理采用什么方法？

【解答】常见的文件存储空间管理有：空闲空间表法、空闲块链接法、位示图法、空闲块成组链接法。

空闲空间表法：所有连续的空闲盘块在表中占有一项，其中每一个空闲表项包括序号、空闲区的第一个盘块号、该区的空闲盘块数及对应的空闲物理块号。利用该表进行盘块的分配和文件删除时盘块的回收。

空闲块链接法：将磁盘上的所有盘块拉成一条链，用一个指针指向第一个空闲块，而各个空闲块中都含有下一个空闲区的块号，最后一块的指针项记为 NULL，表示链尾。分配和释放盘块都在链头进行。

位示图法：利用一串二进制位的值来反映磁盘空间的分配情况，每个盘块都对应一位。若盘块空闲，对应位为 0；若盘块已分配出去，则对应位为 1。

空闲块成组链接法：把所有空闲盘块按固定数量分组，组与组之间形成链接关系，最后一组的块号通常放在内存的一个专用栈中。因而对盘块的分配和回收都在栈中进行。

Linux 系统的存储空间管理采用空闲块成组链接法。

6. 内存管理和外存管理有何异同？

【解答】可以从以下几个方面进行比较：

(1) 主要任务：内存管理的主要任务是为多道程序的运行提供良好的环境，使每到程序都能顺利运行。外存管理的主要任务是为文件提供存储空间。

(2) 基本功能：内存管理的基本功能包含了内存空间的分配、回收、保护、扩充等方面。外存管理的基本功能则只是对外存空间的分配和回收。

(3) 分配方式：内存和外存管理都可采用连续或离散分配方式，且都以离散分配方式为主。

(4) 分配算法：对于连续分配方式，内存与外存管理中的分配和回收算法类似，主要有首次适应算法、循环首次适应算法等。在离散分配方式中，两者采用的机制不同，内存管理只要利用页表或段表；在外存管理中，则主要利用文件分配表。

(5) 分配单位：内存以字节为单位，外存则以盘块为单位分配。

7. 为什么要打开文件？叙述在 linux 文件系统，打开文件/home/user01/myfile 的过程？

【解答】当用户要求对一个文件实施多次读/写或其他操作时，每次都要从检索目录开始。为了避免多次重复地检索目录，在大多数 OS 中都引入了“打开”（open）这一文件系统调用，当用户第一次请求对某文件进行操作时，先利用 open 系统调用将该文件打开。

在 UNIX 文件系统，打开文件/home/user01/myfile 的过程四步：

(1) 检索目录

核心先调用检索目录过程 `namei` 从根目录或从当前目录开始，沿目录树查找指名文件的索引结点。在查找时，利用线性检索法，将文件路径名中的各分量名，与相应目录文件中的文件名逐一进行比较。若未找到该文件名，或者该文件不允许存取，便做出错处理；否则，进入第二步。

(2) 分配内存索引结点

如果该文件已被其他用户打开，此时只需对在第一步中所找到的 `i` 结点，执行其引用计数加 1 的操作；否则，应为被打开文件分配一个内存 `i` 结点，并调用磁盘读过程将磁盘 `i` 结点的内容拷贝到内存 `i` 结点中，并设置 `i.count` 为 1。

(3) 分配文件表

这是指为已打开的文件分配一个文件表项，使文件表项中的 `f.node` 指向内存索引结点。通常还将读写指针 `f.offset` 置为 0，以表示从头开始读 / 写此文件；置读写标志 `f.flag`，及将文件的引用计数 `f.count` 加 1，并记入该表项的首址 `fp`。

(4) 分配用户文件描述表项

在用户文件描述表中取得一空表项。若成功，便将 `fp` 填入该表项中，并把该表项的序号 `fd` 作为文件描述符，写入调用进程的 U 区中。

8. 在 Linux 系统中，一个盘块大小为 1KB，每个盘块号占 4B，则一个进程要访问一个相对于文件开始的偏移量为 263168B 处的数据时，计算是直接还是索引访问，几级索引？

【解答】一次间接。

$$263168B / 1024B = 257$$

即该地址在第 257 块。

$$\text{块内偏移量为：} 263168 - 1024 * 257 = 0$$

$$\text{因 } 10 < 257 < 266$$

故为一次间接。

9. 有一个 UNIX 文件的权限为：`rwxr_x_ _`，该文件的文件主为 `uid=12, gid=1`，有一用户 B 的 `uid=6, gid=1`，问：系统是否允许用户 B 执行该文件？为什么？

【解答】允许。

因为 UID 为用户标识符。说明用户 B 与该文件主为同组用户。

10. 说明 Linux 系统磁盘的结构及各部分的功能。

【解答】(1)

0 号块	1 号块	2 号块...K 号块 (索引结点区)	文件区
------	------	---------------------	-----

(2) 0 号块存放引导程序，是引导区，1 号块是超级块，也叫超级块，存放文件卷的管理信息，索引节点区存放各文件的索引结点；文件区存放文件的内容。

11. 磁盘容量为 1.2GB 的硬盘，假设盘块大小为 1KB，每个盘块号在 FAT 表区占 4B，你作为磁盘的设计人员，应设 FAT 区为多长？

【解答】 $1.2GB = 1.2M * 1KB$

$$\text{即 } 1.2GB / 1KB = 1.2M$$

磁盘由 1.2 个磁盘块组成。

$1.2 \times 4B = 4.8MB$

所以磁盘的 FAT 表区长度应为 4.8MB

12. 设在 linux 中有一进程 P, P 中有一操作需要访问一文件中偏移量为 14000 处的数据; 试问 linux 如何实现地址变换?

【解答】(1) 核心将 14000 换为逻辑块号 13 及块内偏移量 688;

(2) 判断, 因 $10 < 13 < 266$, 故为一次间址;

(3) 从 `i.add(10)` 中取得盘块号, 设为 x;

(4) 调用 `bread` 过程读 x 盘块;

(5) 在一次间址中的文件逻辑块号为 3 (从 0 编);

(6) 从中得实际块号, 设为 y;

(7) 则该块中的 688B 即为所求。

13. 多操作系统中提供了文件重命名功能, 它能赋予文件一个新名字。若进行文件复制, 并给复制文件起一个名字, 然后删除旧文件, 也能达到给文件重命名的目的。试问这两种方法在实现上有何不同?

【解答】使用文件重命名功能时, 用户必须提供两个参数: 旧文件名, 新文件名。实现该功能时, 系统使用旧文件名查找文件目录, 若找到旧文件名所对应的目录表目, 则将目录表中文件名字段对应的值改为新文件名值。从实现过程看, 文件重命名功能完成的工作是修改目录表中的文件名字段, 除文件名外, 文件的其它特性都未改变。

在后一种实现方法中, 先进行文件复制并给复制文件起一个新名, 此时系统完成了一次物理文件的复制操作, 然后删除旧文件。虽然这样也能实现给文件重命名的目的, 但其实现过程比前一种方式复杂, 并且新文件与旧文件的物理存放地址肯定不同。

14. 在利用 `link` 系统调用进行连接时, 如果 `path1=/usr/zhang/ff.c`, `path2=/usr/li/message/ff.c`, 试画出此时的树型目录图。

【解答】在树型文件系统中, 每一个文件都有一路径名。例如, 用户 `zhang` 的文件 `ff.c` 的路径名为 `/usr/zhang/ff.c`。如果用户 `li` 要共享此文件, 可利用系统调用

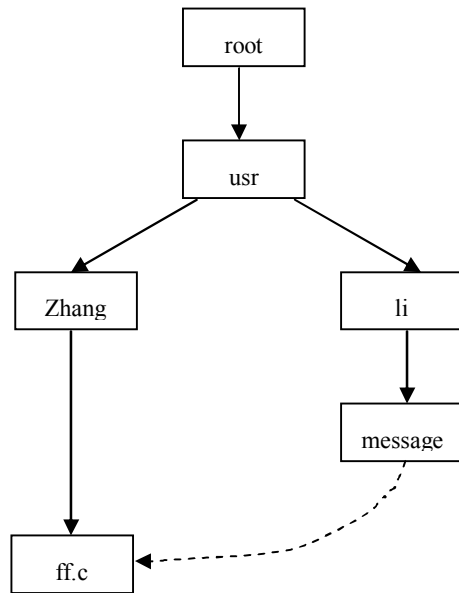
`link(path1,path2)`

与文件 `ff.c` 建立连接。其中参数

`path1=/usr/zhang/ff.c`

`path2=/usr/li/message/ff.c`,

即在 `li` 的 `message` 目录文件中创建一个新目录项。建立连接后的树型目录结构图如下:



15. 在实现文件共享时, 如果不先进行连接, 会产生什么后果?

【解答】建立连接的目的, 一方面是为共享该文件的用户增加一新的文件路径名, 另一方面是让系统知道有多少个用户需要使用该文件。在每次连接时, 该文件的索引结点访问计数会增加 1。共享该文件时, 若无建立连接的操作, 系统就无法知道是否还有用户需要该文件, 其后果可能是: 当用户要使用该文件时, 发现该文件已被删除; 或者在用户使用该文件期间, 文件突然被删除。