

盘调度算法，能分析磁盘相关的性能等。这些都是综合题易考查的内容。

## 4.1 文件系统基础

在学习本节时，请读者思考以下问题：

- 1) 什么是文件？什么是文件系统？
- 2) 文件系统要完成哪些功能？

本节内容较为抽象，对于初学者，推荐配合相关教材的相关章节进行学习。学习过程中要注意区分文件的逻辑结构和物理结构，不要把二者混为一谈。在读者的学习过程中，可尝试以上面的两个问题为线索，构建整个文件系统的概念，先思考有什么方法可以实现文件的共享和保护，再将自己的方法与书上的方法相比较。

在前面的学习中，曾经提醒过读者不要忽略对基本概念的理解。操作系统这门课程在统考中算是比较容易得分的一门课程，从历年的情况来看，大部分同学对进程管理、内存管理有较好的掌握，但对于文件管理及下一章的 I/O 管理，往往理解不太深入，记忆不太牢固，在考试中，即使面对一些关于本章内容的基本问题也容易失分，这十分可惜。主要原因还是对概念的理解不够全面和透彻，希望各位读者能够关注这个问题。

### 4.1.1 文件的概念

#### 1. 文件的定义

文件（File）是操作系统中的一个重要概念。文件是以计算机硬盘为载体的存储在计算机上的信息集合，文件可以是文本文档、图片、程序等。在系统运行时，计算机以进程为基本单位进行资源的调度和分配；而在用户进行的输入、输出中，则以文件为基本单位。大多数应用程序的输入都是通过文件来实现的，其输出也都保存在文件中，以便信息的长期存储及将来的访问。当用户将文件用于应用程序的输入、输出时，还希望可以访问文件、修改文件和保存文件等，实现对文件的维护管理，这就需要系统提供一个文件管理系统，操作系统中的文件系统（File System）就是用于实现用户的这些管理要求的。

要清晰地理解文件的概念，就要了解文件究竟由哪些东西组成。

首先，文件中肯定包括一块存储空间，更准确地说，是存储空间中的数据；其次，由于操作系统要管理成千上万的数据，因此必定需要对这些数据进行划分，然后贴上“标签”，以便于分类和索引，所以文件必定包含分类和索引的信息；最后，不同的用户拥有对数据的不同访问权限，因此文件中一定包含一些关于访问权限的信息。

再举生活中的一个直观例子来类比文件，相信读者了解这个例子后会更深入地了解文件。这个例子就是图书馆中的书，可以认为，计算机中的一个文件相当于图书馆中的一本书，操作系统管理文件，相当于图书管理员管理图书馆中的书。

首先，一本书的主体一定是书中的内容，相当于文件中的数据；其次，不同类别的书需要放在不同的书库，然后加上编号，再把编号登记在图书管理系统中，方便读者查阅，相当于文件的分类和查找；最后，有些已经绝版或价格比较高的外文书籍，只能借给 VIP 会员或权限比较高的其他读者，而有些普通的书籍可供任何人借阅，这就是文件中的访问权限。

所举的例子与实际操作系统中的情形并不绝对等价，读者应能找出类比中的不严谨之处，但对于某些关键的属性，图书馆管理图书和操作系统管理文件的思想却有相一致的地方，因此通过

这种类比可使初学者快速认识陌生的概念。

从用户的角度看，文件系统是操作系统的重要部分之一。用户关心的是如何命名、分类和查找文件，如何保证文件数据的安全性及对文件可以进行哪些操作等。而对其中的细节，如文件如何存储在辅存上、如何管理文件辅存区域等关心甚少。

文件系统提供了与二级存储相关的资源的抽象，让用户能在不了解文件的各种属性、文件存储介质的特征及文件在存储介质上的具体位置等情况下，方便快捷地使用文件。

用户通过文件系统建立文件，提供应用程序的输入、输出，对资源进行管理。首先了解文件的结构，我们通过自底向上的方式来定义。

1) 数据项。数据项是文件系统中最最低级的数据组织形式，可分为以下两种类型：

- 基本数据项。用于描述一个对象的某种属性的一个值，如姓名、日期或证件号等，是数据中可命名的最小逻辑数据单位，即原子数据。
- 组合数据项。由多个基本数据项组成。

2) 记录。记录是一组相关数据项的集合，用于描述一个对象在某方面的属性，如一名考生的报名记录包括考生姓名、出生日期、报考学校代号、身份证号等一系列域。

3) 文件。文件是指由创建者所定义的一组相关信息的集合，逻辑上可分为有结构文件和无结构文件两种。在有结构文件中，文件由一组相似的记录组成，如报考某学校的所有考生的报考信息记录，又称记录式文件；而无结构文件则被视为一个字符流，比如一个二进制文件或字符文件，又称流式文件。

虽然上面给出了结构化的表述，但实际上关于文件并无严格的定义。在操作系统中，通常将程序和数据组织成文件。文件可以是数字、字母或二进制代码，基本访问单元可以是字节、行或记录。文件可以长期存储于硬盘或其他二级存储器中，允许可控制的进程间共享访问，能够被组织成复杂的结构。

## 2. 文件的属性

文件具有一定的属性，系统不同，属性也会有所不同，但通常都包括如下属性。

- 1) 名称。文件名称唯一，以容易读取的形式保存。
- 2) 标识符。标识文件系统内文件的唯一标签，通常为数字，是对人不可读的一种内部名称。
- 3) 类型。被支持不同类型的文件系统所使用。
- 4) 位置。指向设备和设备上文件的指针。
- 5) 大小。文件当前大小（用字节、字或块表示），也可包含文件允许的最大值。
- 6) 保护。对文件进行保护的访问控制信息。
- 7) 时间、日期和用户标识。文件创建、上次修改和上次访问的相关信息，用于保护和跟踪文件的使用。

所有文件的信息都保存在目录结构中，而目录结构保存在外存上。文件信息在需要时才调入内存。通常，目录条目包括文件名称及其唯一的标识符，而标识符定位其他属性的信息。

## 3. 文件的基本操作

文件属于抽象数据类型。为了恰当地定义文件，需要考虑有关文件的操作。操作系统提供系统调用，它对文件进行创建、写、读、重定位、删除和截断等操作。

- 1) 创建文件。创建文件有两个必要步骤：一是在文件系统中为文件找到空间；二是在目录中为新文件创建条目，该条目记录文件名称、在文件系统中的位置及其他可能的信息。
- 2) 写文件。为了写文件，执行一个系统调用，指明文件名称和要写入文件的内容。对于给

定文件名称，系统搜索目录以查找文件位置。系统必须为该文件维护一个写位置的指针。每当发生写操作时，便更新写指针。

3) 读文件。为了读文件，执行一个系统调用，指明文件名称和要读入文件块的内存位置。

同样，需要搜索目录以找到相关目录项，系统维护一个读位置的指针。每当发生读操作时，更新读指针。一个进程通常只对一个文件读或写，因此当前操作位置可作为每个进程当前文件位置的指针。由于读和写操作都使用同一指针，因此节省了空间，也降低了系统复杂度。

4) 文件重定位(文件寻址)。按某条件搜索目录，将当前文件位置设为给定值，并且不会读、写文件。

5) 删除文件。先从目录中找到要删除文件的目录项，使之成为空项，然后回收该文件所占用的存储空间。

6) 截断文件。允许文件所有属性不变，并删除文件内容，即将其长度设为 0 并释放其空间。

这 6 个基本操作可以组合起来执行其他文件操作。例如，一个文件的复制，可以创建新文件、从旧文件读出并写入新文件。

#### 4. 文件的打开与关闭

因为许多文件操作都涉及为给定文件搜索相关目录条目，因此许多系统要求在首次使用文件时，使用系统调用 open 将指明文件的属性（包括该文件在外存上的物理位置）从外存复制到内存打开文件表的一个表目中，并将该表目的编号（也称索引）返回给用户。操作系统维护一个包含所有打开文件信息的表（打开文件表，open-file table）。当用户需要一个文件操作时，可通过该表的一个索引指定文件，因此省略了搜索环节。当文件不再使用时，进程可以关闭它，操作系统从打开文件表中删除这一条目。

大部分操作系统要求在文件使用之前就被显式地打开。操作 open 会根据文件名搜索目录，并将目录条目复制到打开文件表。若调用 open 的请求（创建、只读、读写、添加等）得到允许，则进程就可打开文件，而 open 通常返回一个指向打开文件表中的一个条目的指针。通过使用该指针（而非文件名）进行所有 I/O 操作，以简化步骤并节省资源。

注意，在 open 调用完成后，操作系统对该文件的任何操作都不再需要文件名，而只需要 open 调用返回的指针。

整个系统表包含进程相关信息，如文件在磁盘的位置、访问日期和大小。一个进程打开一个文件，系统打开文件表就会为打开的文件增加相应的条目。当另一个进程执行 open 时，只不过是在其进程打开表中增加一个条目，并指向整个系统表的相应条目。通常，系统打开文件表的每个文件时，还用一个文件打开计数器（Open Count），以记录多少进程打开了该文件。每个关闭操作 close 使 count 递减，当打开计数器为 0 时，表示该文件不再被使用，系统将回收分配给该文件的内存空间等资源。若文件被修改过，则将文件写回外存，并将系统打开文件表中的相应条目删除，最后释放文件的文件控制块（File Control Block，FCB）。

每个打开文件都有如下关联信息：

- 文件指针。系统跟踪上次的读写位置作为当前文件位置的指针，这种指针对打开文件的某个进程来说是唯一的，因此必须与磁盘文件属性分开保存。
- 文件打开计数。文件关闭时，操作系统必须重用其打开文件表条目，否则表内空间会不够用。因为多个进程可能打开同一个文件，所以系统在删除打开文件条目之前，必须等待最后一个进程关闭文件。计数器跟踪打开和关闭的数量，计数为 0 时，系统关闭文件，删除该条目。

- 文件磁盘位置。绝大多数文件操作都要求系统修改文件数据。该信息保存在内存中，以免为每个操作都从磁盘中读取。
- 访问权限。每个进程打开文件都需要有一个访问模式（创建、只读、读写、添加等）。该信息保存在进程的打开文件表中，以便操作系统能够允许或拒绝之后的 I/O 请求。

### 4.1.2 文件的逻辑结构

文件的逻辑结构是从用户观点出发看到的文件的组织形式。文件的物理结构（又称文件的存储结构，见 4.2.1 节）是从实现观点出发看到的文件在外存上的存储组织形式。文件的逻辑结构与存储介质特性无关，但文件的物理结构与存储介质的特性有很大关系。文件的逻辑结构实际上是指在文件的内部，数据逻辑上是如何组织起来的。

按逻辑结构，文件可划分为无结构文件和有结构文件两种。

#### 1. 无结构文件（流式文件）

无结构文件是最简单的文件组织形式。无结构文件将数据按顺序组织成记录并积累、保存，它是有序相关信息项的集合，以字节（Byte）为单位。由于无结构文件没有结构，因而对记录的访问只能通过穷举搜索的方式，因此这种文件形式对大多数应用不适用。但字符流的无结构文件管理简单，用户可以方便地对其进行操作。所以，那些对基本信息单位操作不多的文件较适于采用字符流的无结构方式，如源程序文件、目标代码文件等。

#### 2. 有结构文件（记录式文件）

有结构文件按记录的组织形式可以分为如下几种：

- 1) 顺序文件。文件中的记录一个接一个地顺序排列，记录通常是定长的，可以顺序存储或以链表形式存储，在访问时需要顺序搜索文件。顺序文件有以下两种结构：第一种是串结构，记录之间的顺序与关键字无关。通常的办法是由时间决定，即按存入时间的先后排列，最先存入的记录作为第 1 条记录，其次存入的为第 2 条记录，以此类推。第二种是顺序结构，指文件中的所有记录按关键字顺序排列。

在对记录进行批量操作，即每次要读或写一大批记录时，顺序文件的效率是所有逻辑文件中最高的；此外，也只有顺序文件才能存储在磁带上，并能有效地工作，但顺序文件对查找、修改、增加或删除单条记录的操作比较困难。

- 2) 索引文件。索引文件示意图如图 4.1 所示。对于定长记录文件，要查找第  $i$  条记录，可直接根据下式计算得到第  $i$  条记录相对于第 1 条记录的地址：

$$A_i = i \times L$$

然而，对于可变长记录的文件，要查找第  $i$  条记录，必须顺序地查找前  $i - 1$  条记录，从而获得相应记录的长度  $L$ ，进而按下式计算出第  $i$  条记录的首址：

$$A_i = \sum_{j=0}^{i-1} L_j + 1$$

注意：假定每条记录前用一个字节指明该记录的长度。

变长记录文件只能顺序查找，系统开销较大。为此，可以建立一张索引表以加快检索速度，索引表本身是定长记录的顺序文件。在记录很多或访问要求高的文件中，需要引入索引以提供有效的访问。实际上，通过索引可以成百上千倍地提高访问速度。

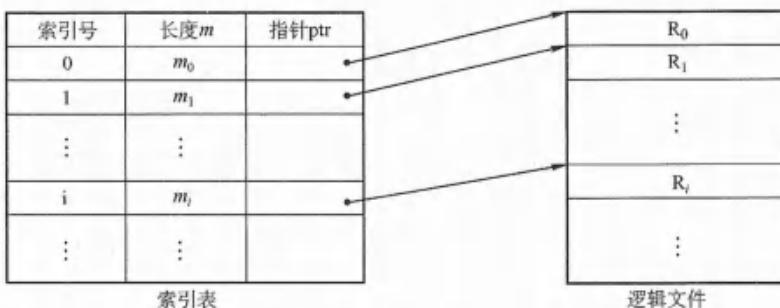


图 4.1 索引文件示意图

3) 索引顺序文件。索引顺序文件是顺序和索引两种组织形式的结合。索引顺序文件将顺序文件中的所有记录分为若干组，为顺序文件建立一张索引表，在索引表中为每组中的第一条记录建立一个索引项，其中含有该记录的关键字值和指向该记录的指针。

如图 4.2 所示，主文件名包含姓名和其他数据项。姓名为关键字，索引表中为每组的第一条记录（不是每条记录）的关键字值，用指针指向主文件中该记录的起始位置。索引表只包含关键字和指针两个数据项，所有姓名关键字递增排列。主文件中记录分组排列，同一个组中的关键字可以无序，但组与组之间的关键字必须有序。查找一条记录时，首先通过索引表找到其所在的组，然后在该组中使用顺序查找，就能很快地找到记录。

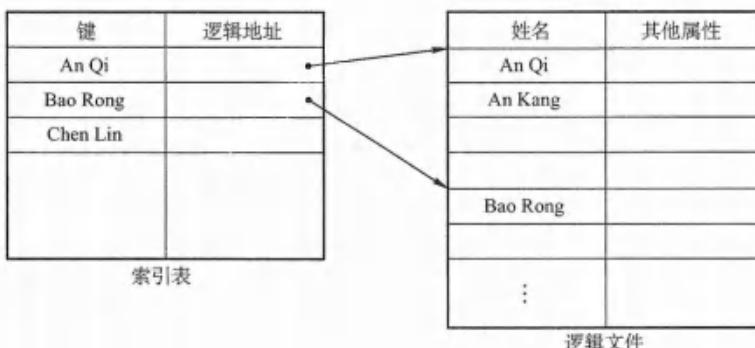


图 4.2 索引顺序文件示意图

对于含有  $N$  条记录的顺序文件，查找某关键字值的记录时，平均需要查找  $N/2$  次。在索引顺序文件中，假设  $N$  条记录分为  $\sqrt{N}$  组，索引表中有  $\sqrt{N}$  个表项，每组有  $\sqrt{N}$  条记录，在查找某关键字值的记录时，先顺序查找索引表，需要查找  $\sqrt{N}/2$  次，然后在主文件中对应的组中顺序查找，也需要查找  $\sqrt{N}/2$  次，因此共需查找  $\sqrt{N}/2 + \sqrt{N}/2 = \sqrt{N}$  次。显然，索引顺序文件提高了查找效率，若记录数很多，则可采用两级或多级索引。

索引文件和索引顺序文件都提高了存取的速度，但因为配置索引表而增加了存储空间。

4) 直接文件或散列文件 (Hash File)。给定记录的键值或通过散列函数转换的键值直接决定记录的物理地址。这种映射结构不同于顺序文件或索引文件，没有顺序的特性。

散列文件有很高的存取速度，但是会引起冲突，即不同关键字的散列函数值相同。

复习了数据结构的读者读到这里时，会有这样的感觉：有结构文件逻辑上的组织，是为在文件中查找数据服务的（顺序查找、索引查找、索引顺序查找、哈希查找）。

### 4.1.3 目录结构

与文件管理系统和文件集合相关联的是文件目录，它包含有关文件的信息如属性、位置和所有权等，这些信息主要由操作系统进行管理。首先我们来看目录管理的基本要求：从用户的角度看，目录在用户（应用程序）所需要的文件名和文件之间提供一种映射，所以目录管理要实现“按名存取”；目录存取的效率直接影响到系统的性能，所以要提高对目录的检索速度；在共享系统中，目录还需要提供用于控制访问文件的信息。此外，文件允许重名也是用户的合理和必然要求，目录管理通过树形结构来解决和实现。

前面介绍了文件内部的逻辑结构，下面介绍多个文件之间在逻辑上是如何组织的，这实际上是文件“外部”的逻辑结构的问题。

#### 1. 文件控制块和索引结点

与进程管理一样，为实现目录管理，操作系统中引入了文件控制块的数据结构。

1) 文件控制块。文件控制块 (FCB) 是用来存放控制文件需要的各种信息的数据结构，以实现“按名存取”。FCB 的有序集合称为文件目录，一个 FCB 就是一个文件目录项。为了创建一个新文件，系统将分配一个 FCB 并存放在文件目录中，成为目录项。

FCB 主要包含以下信息：

- 基本信息，如文件名、文件的物理位置、文件的逻辑结构、文件的物理结构等。
- 存取控制信息，如文件存取权限等。
- 使用信息，如文件建立时间、修改时间等。

2) 索引结点。在检索目录文件的过程中，只用到了文件名，仅当找到一个目录项（查找文件名与目录项中文件名匹配）时，才需要从该目录项中读出该文件的物理地址。也就是说，在检索目录时，文件的其他描述信息不会用到，也不需要调入内存。因此，有的系统（如 UNIX，见表 4.1）采用了文件名和文件描述信息分开的方法，文件描述信息单独形成一个称为索引结点的数据结构，简称 i 结点。在文件目录中的每个目录项仅由文件名和指向该文件所对应的 i 结点的指针构成。

表 4.1 UNIX 的文件目录结构

文件名	索引结点编号
文件名 1	
文件名 2	
:	
:	

一个 FCB 的大小是 64B，盘块大小是 1KB，因此在每个盘块中可以存放 16 个 FCB（注意，FCB 必须连续存放）。而在 UNIX 系统中，一个目录项仅占 16B，其中 14B 是文件名，2B 是 i 结点指针。在 1KB 的盘块中可存放 64 个目录项。这样，就可使查找文件时的平均启动磁盘次数减少到原来的 1/4，大大节省了系统开销。

存放在磁盘上的索引结点称为磁盘索引结点，UNIX 中的每个文件都有一个唯一的磁盘索引结点，主要包括以下几个方面：

- 文件主标识符，拥有该文件的个人或小组的标识符。
- 文件类型，包括普通文件、目录文件或特别文件。
- 文件存取权限，各类用户对该文件的存取权限。
- 文件物理地址，每个索引结点中含有 13 个地址项，即  $iaddr(0) \sim iaddr(12)$ ，它们以直接或间接方式给出数据文件所在盘块的编号。
- 文件长度，以字节为单位。
- 文件链接计数，在本文件系统中所有指向该文件的文件名的指针计数。

- 文件存取时间，本文件最近被进程存取的时间、最近被修改的时间及索引结点最近被修改的时间。

文件被打开时，磁盘索引结点复制到内存的索引结点中，以便于使用。在内存索引结点中又增加了以下内容：

- 索引结点编号，用于标识内存索引结点。
- 状态，指示 i 结点是否上锁或被修改。
- 访问计数，每当有一进程要访问此 i 结点时，计数加 1，访问结束减 1。
- 逻辑设备号，文件所属文件系统的逻辑设备号。
- 链接指针，设置分别指向空闲链表和散列队列的指针。

FCB 或索引结点相当于图书馆中图书的索书号，我们可以在图书馆网站上找到图书的索书号，然后根据索书号找到想要的书本。

## 2. 目录结构

在理解一个文件系统的需求前，我们首先考虑在目录这个层次上所需要执行的操作，这有助于后面文件系统的整体理解。

- 搜索。当用户使用一个文件时，需要搜索目录，以找到该文件的对应目录项。
- 创建文件。当创建一个新文件时，需要在目录中增加一个目录项。
- 删除文件。当删除一个文件时，需要在目录中删除相应的目录项。
- 显示目录。用户可以请求显示目录的内容，如显示该用户目录中的所有文件及属性。
- 修改目录。某些文件属性保存在目录中，因而这些属性的变化需要改变相应的目录项。

操作时，考虑以下几种目录结构：

- 单级目录结构。在整个文件系统中只建立一张目录表，每个文件占一个目录项，如图 4.3 所示。

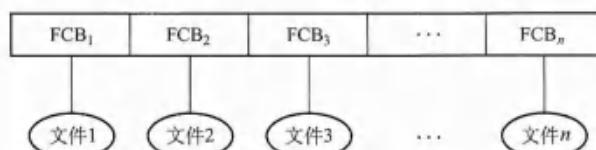


图 4.3 单级目录结构

当访问一个文件时，先按文件名在该目录中查找到相应的 FCB，经合法性检查后执行相应操作。当建立一个新文件时，必须先检索所有目录项以确保没有“重名”的情况，然后在该目录中增设一项，把 FCB 的全部信息保存在该项中。当删除一个文件时，先从该目录中找到该文件的目录项，回收该文件所占用的存储空间，然后清除该目录项。

单级目录结构实现了“按名存取”，但是存在查找速度慢、文件不允许重名、不便于文件共享等缺点，而且对于多用户的操作系统显然是不适用的。

- 两级目录结构。单级目录很容易造成文件名称的混淆，因此可以考虑采用两级方案，将文件目录分成主文件目录（Master File Directory，MFD）和用户文件目录（User File Directory，UFD）两级，如图 4.4 所示。

主文件目录项记录用户名及相应用户文件目录所在的存储位置。用户文件目录项记录该用户文件的 FCB 信息。当某用户欲对其文件进行访问时，只需搜索该用户对应的 UFD，这既解决了不同用户文件的“重名”问题，又在一定程度上保证了文件的安全。

两级目录结构可以解决多用户之间的文件重名问题，文件系统可以在目录上实现访问限制。但是两级目录结构缺乏灵活性，不能对文件分类。

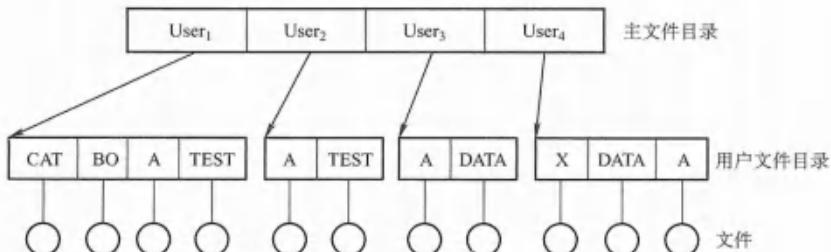


图 4.4 两级目录结构

3) 多级目录结构(树形目录结构)。将两级目录结构的层次关系加以推广，就形成了多级目录结构，即树形目录结构，如图 4.5 所示。

用户要访问某个文件时，用文件的路径名标识文件，文件路径名是个字符串，由从根目录出发到所找文件通路上所有目录名与数据文件名用分隔符“/”链接而成。从根目录出发的路径称为绝对路径。当层次较多时，每次从根目录查询会浪费时间，于是加入了当前目录(又称工作目录)，进程对各文件的访问都是相对于当前目录进行的。当用户要访问某个文件时，使用相对路径标识文件，相对路径由从当前目录出发到所找文件通路上所有目录名与数据文件名用分隔符“/”链接而成。

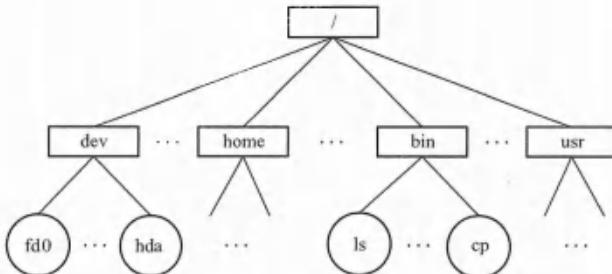


图 4.5 树形目录结构

图 4.5 是 Linux 操作系统的目录结构，“/dev/hda”就是一个绝对路径。若当前目录为“/bin”，则“./ls”就是一个相对路径，其中符号“.”表示当前工作目录。

通常，每个用户都有各自的“当前目录”，登录后自动进入该用户的“当前目录”。操作系统提供一条专门的系统调用，供用户随时改变“当前目录”。例如，在 UNIX 系统中，“/etc/passwd”文件就包含有用户登录时默认的“当前目录”，可用 cd 命令改变“当前目录”。

树形目录结构可以很方便地对文件进行分类，层次结构清晰，也能够更有效地进行文件的管理和保护。但是，在树形目录中查找一个文件时，需要按路径名逐级访问中间结点，这就增加了磁盘访问次数，无疑将影响查询速度。

4) 无环图目录结构。树形目录结构能便于实现文件分类，但不便于实现文件共享，为此在树形目录结构的基础上增加了一些指向同一结点的有向边，使整个目录成为一个有向无环图。引入无环图目录结构是为了实现文件共享，如图 4.6 所示。

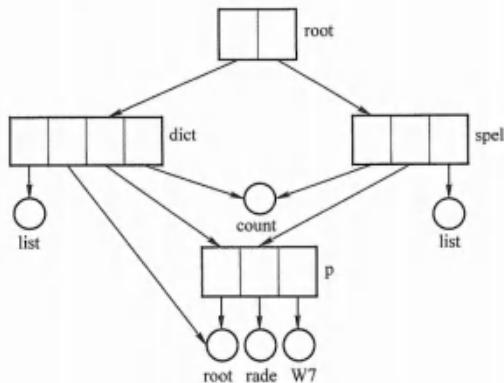


图 4.6 无环图目录结构

当某用户要求删除一个共享结点时，若系统只是简单地将它删除，则当另一共享用户需要访问时，会因无法找到这个文件而发生错误。为此，可为每个共享结点设置一个共享计数器，每当图中增加对该结点的共享链时，计数器加 1；每当某用户提出删除该结点时，计数器减 1。仅当共享计数器为 0 时，才真正删除该结点，否则仅删除请求用户的共享链。共享文件（或目录）不同于文件拷贝（副本）。若有两个文件拷贝，则每个程序员看到的是拷贝而不是原件；然而，若一个文件被修改，则另一个程序员的拷贝不会改变。对于共享文件，只存在一个真正的文件，任何改变都会为其他用户所见。

无环图目录结构方便地实现了文件的共享，但使得系统的管理变得更加复杂。

#### 4.1.4 文件共享

文件共享使多个用户（进程）共享同一个文件，系统中只需保留该文件的一个副本。若系统不能提供共享功能，则每个需要该文件的用户都要有各自的副本，会造成对存储空间的极大浪费。随着计算机技术的发展，文件共享的范围已由单机系统发展到多机系统，进而通过网络扩展到全球。这些文件的分享是通过分布式文件系统、远程文件系统、分布式信息系统实现的。这些系统允许多个客户通过 C/S 模型共享网络中的服务器文件。

现代常用的两种文件共享方法如下。

##### 1. 基于索引结点的共享方式（硬链接）

在树形结构的目录中，当有两个或多个用户要共享一个子目录或文件时，必须将共享文件或子目录链接到两个或多个用户的目录中，才能方便地找到该文件，如图 4.7 所示。

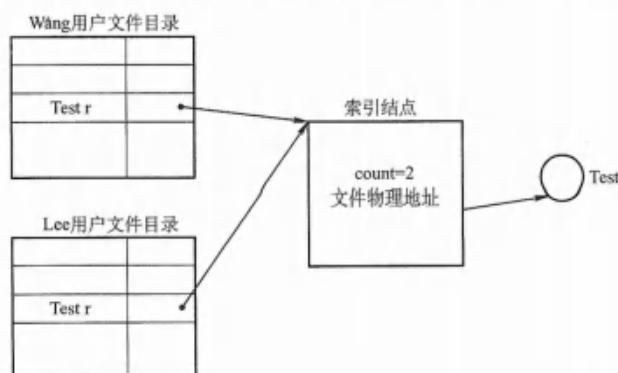


图 4.7 基于索引结点的共享方式

在这种共享方式中，诸如文件的物理地址及其他文件属性等信息，不再放在目录项中，而放在索引结点中。在文件目录中只设置文件名及指向相应索引结点的指针。在索引结点中还应有一个链接计数 count，用于表示链接到本索引结点（即文件）上的用户目录项的数目。当  $count = 2$  时，表示有两个用户目录项链接到本文件上，或者说有两个用户共享此文件。

用户 A 创建一个新文件时，它便是该文件的所有者，此时将 count 置为 1。用户 B 要共享此文件时，在用户 B 的目录中增加一个目录项，并设置一个指针指向该文件的索引结点。此时，文件主仍然是用户 A， $count = 2$ 。用户 A 不再需要此文件，不能将文件直接删除。因为若删除了该文件，则必然也删除了该文件的索引结点，这样便会使用户 B 的指针悬空，而用户 B 可能正在此文件上执行写操作，此时用户 B 会无法访问到文件。因此用户 A 不能删除此文件，只是将该文件的 count 减 1，然后删除自己目录中的相应目录项。用户 B 仍可以使用该文件。当  $count = 0$  时，表示没有用户使用该文件，系统将负责删除该文件。如图 4.8 给出了用户 B 链接到文件上的前、后情况。

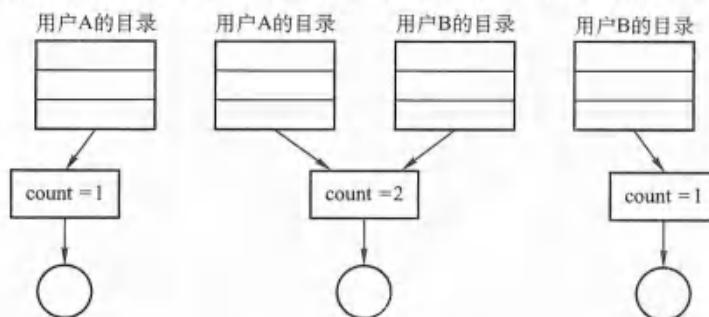


图 4.8 文件共享中的链接计数

## 2. 利用符号链实现文件共享（软链接）

为使用户 B 能共享用户 A 的一个文件 F，可以由系统创建一个 LINK 类型的新文件，也取名为 F，并将文件 F 写入用户 B 的目录中，以实现用户 B 的目录与文件 F 的链接。在新文件中只包含被链接文件 F 的路径名。这样的链接方法被称为符号链接。

新文件中的路径名只被视为符号链，当用户 B 要访问被链接的文件 F 且正要读 LINK 类新文件时，操作系统根据新文件中的路径名去读该文件，从而实现用户 B 对文件 F 的共享。

在利用符号链方式实现文件共享时，只有文件的拥有者才拥有指向其索引结点的指针。而共享该文件的其他用户只有该文件的路径名，并不拥有指向其索引结点的指针。这样，也就不会发生在文件主删除一个共享文件后留下一个悬空指针的情况。当文件的拥有者把一个共享文件删除后，其他用户通过符号链去访问它时，会出现访问失败，于是将符号链删除，此时不会产生任何影响。当然，利用符号链实现文件共享仍然存在问题。例如，一个文件采用符号链方式共享，当文件拥有者将其删除，而在共享的其他用户使用其符号链接访问该文件之前，又有人在同一路径下创建了另一个具有同样名称的文件，则该符号链将仍然有效，但访问的文件已经改变，从而导致错误。

在符号链的共享方式中，当其他用户读共享文件时，需要根据文件路径名逐个地查找目录，直至找到该文件的索引结点。因此，每次访问时，都可能要多次地读盘，使得访问文件的开销变大并增加了启动磁盘的频率。此外，符号链的索引结点也要耗费一定的磁盘空间。

符号链方式有一个很大的优点，即网络共享只需提供该文件所在机器的网络地址及该机器中的文件路径。

上述两种链接方式都存在一个共同的问题，即每个共享文件都有几个文件名。换言之，每增加一条链接，就增加一个文件名。这实质上是每个用户都使用自己的路径名去访问共享文件。当

我们试图去遍历整个文件系统时，将会多次遍历到该共享文件。

硬链接和软链接都是文件系统中的静态共享方法，在文件系统中还存在着另外的共享需求，即两个进程同时对同一个文件进行操作，这样的共享称为动态共享。

可以这样说：文件共享，“软”“硬”兼施。硬链接就是多个指针指向一个索引结点，保证只要还有一个指针指向索引结点，索引结点就不能删除；软链接就是把到达共享文件的路径记录下来，当要访问文件时，根据路径寻找文件。可以想象，硬链接的查找速度要比软链接的快。

### 4.1.5 文件保护

为了防止文件共享可能会导致文件被破坏或未经核准的用户修改文件，文件系统必须控制用户对文件的存取，即解决对文件的读、写、执行的许可问题。为此，必须在文件系统中建立相应的文件保护机制。

文件保护通过口令保护、加密保护和访问控制等方式实现。其中，口令保护和加密保护是为了防止用户文件被他人存取或窃取，而访问控制则用于控制用户对文件的访问方式。

#### 1. 访问类型

对文件的保护可从限制对文件的访问类型中出发。可加以控制的访问类型主要有以下几种。

- 读。从文件中读。
- 写。向文件中写。
- 执行。将文件装入内存并执行。
- 添加。将新信息添加到文件结尾部分。
- 删除。删除文件，释放空间。
- 列表清单。列出文件名和文件属性。

此外还可以对文件的重命名、复制、编辑等加以控制。这些高层的功能可以通过系统程序调用低层系统调用实现。保护可以只在低层提供。例如，复制文件可利用一系列的读请求来完成，这样，具有读访问权限的用户同时也具有了复制和打印权限。

#### 2. 访问控制

解决访问控制最常用的方法是根据用户身份进行控制。而实现基于身份访问的最为普通的方法是，为每个文件和目录增加一个访问控制列表（Access-Control List, ACL），以规定每个用户名及其所允许的访问类型。

这种方法的优点是可以使用复杂的访问方法，缺点是长度无法预计并且可能导致复杂的空间管理，使用精简的访问列表可以解决这个问题。

精简的访问列表采用拥有者、组和其他三种用户类型。

- 1) 拥有者。创建文件的用户。
- 2) 组。一组需要共享文件且具有类似访问的用户。
- 3) 其他。系统内的所有其他用户。

这样，只需用三个域即可列出访问表中这三类用户的访问权限。文件拥有者在创建文件时，说明创建者用户名及所在的组名，系统在创建文件时也将文件主的名字、所属组名列在该文件的FCB中。用户访问该文件时，按照拥有者所拥有的权限访问文件，若用户和拥有者在同一个用户组，则按照同组权限访问，否则只能按其他用户权限访问。UNIX 操作系统即采用此种方法。

口令和密码是另外两种访问控制方法。

口令指用户在建立一个文件时提供一个口令，系统为其建立 FCB 时附上相应口令，同时告

诉允许共享该文件的其他用户。用户请求访问时必须提供相应的口令。这种方法时间和空间的开销不多，缺点是口令直接存在系统内部，不够安全。

密码指用户对文件进行加密，文件被访问时需要使用密钥。这种方法保密性强，节省了存储空间，不过编码和译码要花费一定的时间。

口令和密码都是防止用户文件被他人存取或窃取，并没有控制用户对文件的访问类型。

注意两个问题：

- 1) 现代操作系统常用的文件保护方法是，将访问控制列表与用户、组和其他成员访问控制方案一起组合使用。
- 2) 对于多级目录结构而言，不仅需要保护单个文件，而且需要保护子目录内的文件，即需要提供目录保护机制。目录操作与文件操作并不相同，因此需要不同的保护机制。

#### 4.1.6 本节小结

本节开头提出的问题的参考答案如下。

##### 1) 什么是文件？什么是文件系统？

文件是以计算机硬盘为载体的存储在计算机上的信息集合，它的形式多样，可以是文本文档、图片、程序等。操作系统中负责管理和存储文件信息的软件机构称为文件管理系统，简称文件系统。文件系统由三部分组成：与文件管理有关的软件、被管理文件及实施文件管理所需的数据结构。

##### 2) 文件系统要完成哪些功能？

对于用户而言，文件系统最主要的功能是实现对文件的基本操作，让用户可以按名存储和查找文件，组织成合适的结构，并应当具有基本的文件共享和文件保护功能。对于操作系统本身而言，文件系统还需要管理与磁盘的信息交换，完成文件逻辑结构和物理结构上的变换，组织文件在磁盘上的存放，采取好的文件排放顺序和磁盘调度方法以提升整个系统的性能。

学习到这里时，读者应会有这样的一种体会：现代操作系统的管理思想中，到处能够见到面向对象程序设计的影子。本节我们学习的一个新概念——文件，实质上就是一个抽象数据类型，也就是一种数据结构，若读者在复习操作系统之前已复习完数据结构，则遇到一种新的数据结构时，一定会有这样的意识：要认识它的逻辑结构、物理结构，以及对这种数据结构的操作。本节我们已经学完文件的逻辑结构，下一节将介绍文件的实现，也就是文件的物理结构。操作系统对文件的操作不是本课程关心的问题，我们不去研究。

#### 4.1.7 本节习题精选

##### 一、单项选择题

1. 【2010 统考真题】设置当前工作目录的主要目的是（ ）。
 

A. 节省外存空间	B. 节省内存空间
C. 加快文件的检索速度	D. 加快文件的读/写速度
2. 【2009 统考真题】文件系统中，文件访问控制信息存储的合理位置是（ ）。
 

A. 文件控制块	B. 文件分配表	C. 用户口令表	D. 系统注册表
----------	----------	----------	----------
3. 从用户的观点看，操作系统中引入文件系统的目的是（ ）。
 

A. 保护用户数据	B. 实现对文件的按名存取
C. 实现虚拟存储	D. 保存用户和系统文档及数据
4. 文件系统在创建一个文件时，为它建立一个（ ）。

- A. 文件目录项      B. 目录文件      C. 逻辑结构      D. 逻辑空间
5. 打开文件操作的主要工作是( )。  
A. 把指定文件的目录复制到内存指定的区域  
B. 把指定文件复制到内存指定的区域  
C. 在指定文件所在的存储介质上找到指定文件的目录  
D. 在内存寻找指定的文件
6. UNIX 操作系统中，输入/输出设备视为( )。  
A. 普通文件      B. 目录文件      C. 索引文件      D. 特殊文件
7. 下列说法中，( )属于文件的逻辑结构的范畴。  
A. 连续文件      B. 系统文件      C. 链接文件      D. 流式文件
8. 文件的逻辑结构是为了方便( )而设计的。  
A. 存储介质特性      B. 操作系统的管理方式  
C. 主存容量      D. 用户
9. 索引文件由逻辑文件和( )组成。  
A. 符号表      B. 索引表      C. 交叉访问表      D. 链接表
10. 下列关于索引表的叙述中，( )是正确的。  
A. 索引表中每条记录的索引项可以有多个  
B. 对索引文件存取时，必须先查找索引表  
C. 索引表中含有索引文件的数据及其物理地址  
D. 建立索引的目的之一是减少存储空间
11. 有一个顺序文件含有 10000 条记录，平均查找的记录数为 5000 个，采用索引顺序文件结构，则最好情况下平均只需查找( )次记录。  
A. 1000      B. 10000      C. 100      D. 500
12. 【2012 统考真题】若一个用户进程通过 read 系统调用读取一个磁盘文件中的数据，则下列关于此过程的叙述中，正确的是( )。  
I. 若该文件的数据不在内存，则该进程进入睡眠等待状态  
II. 请求 read 系统调用会导致 CPU 从用户态切换到核心态  
III. read 系统调用的参数应包含文件的名称  
A. 仅 I、II      B. 仅 I、III      C. 仅 II、III      D. I、II 和 III
13. 【2013 统考真题】用户在删除某文件的过程中，操作系统不可能执行的操作是( )。  
A. 删除此文件所在的目录      B. 删除与此文件关联的目录项  
C. 删除与此文件对应的文件控制块      D. 释放与此文件关联的内存缓冲区
14. 一个文件的相对路径名是从( )开始，逐步沿着各级子目录追溯，最后到指定文件的整个通路上所有子目录名组成的一个字符串。  
A. 当前目录      B. 根目录      C. 多级目录      D. 二级目录
15. 目录文件存放的信息是( )。  
A. 某一文件存放的数据信息  
B. 某一文件的文件目录  
C. 该目录中所有数据文件目录  
D. 该目录中所有子目录文件和数据文件的目录
16. FAT32 的文件目录项不包括( )。

- A. 文件名                          B. 文件访问权限说明  
  C. 文件控制块的物理位置        D. 文件所在的物理位置
17. 文件系统采用多级目录结构的目的是( )。  
  A. 减少系统开销    B. 节省存储空间    C. 解决命名冲突    D. 缩短传送时间
18. 若文件系统中有两个文件重名，则不应采用( )。  
  A. 单级目录结构    B. 两级目录结构    C. 树形目录结构    D. 多级目录结构
19. UNIX操作系统中，文件的索引结构放在( )。  
  A. 超级块                          B. 索引结点                          C. 目录项                          D. 空闲块
20. 操作系统为保证未经文件拥有者授权，任何其他用户不能使用该文件，所提供的解决方法是( )。  
  A. 文件保护                          B. 文件保密                          C. 文件转储                          D. 文件共享
21. 【2009统考真题】设文件F1的当前引用计数值为1，先建立文件F1的符号链接（软链接）文件F2，再建立文件F1的硬链接文件F3，然后删除文件F1。此时，文件F2和文件F3的引用计数值分别是( )。  
  A. 0, 1                                  B. 1, 1                                  C. 1, 2                                  D. 2, 1
22. 【2017统考真题】若文件f1的硬链接为f2，两个进程分别打开f1和f2，获得对应的文件描述符为fd1和fd2，则下列叙述中正确的是( )。  
 I. f1和f2的读写指针位置保持相同  
 II. f1和f2共享同一个内存索引结点  
 III. fd1和fd2分别指向各自的用户打开文件表中的一项  
  A. 仅III                                  B. 仅II、III                          C. 仅I、II                                  D. I、II和III
23. 在文件系统中，以下不属于文件保护的方法是( )。  
  A. 口令    B. 存取控制  
  C. 用户权限表                                  D. 读写之后使用关闭命令
24. 对一个文件的访问，常由( )共同限制。  
  A. 用户访问权限和文件属性                          B. 用户访问权限和用户优先级  
  C. 优先级和文件属性                                  D. 文件属性和口令
25. 加密保护和访问控制两种机制相比，( )。  
  A. 加密保护机制的灵活性更好                          B. 访问控制机制的安全性更高  
  C. 加密保护机制必须由系统实现                          D. 访问控制机制必须由系统实现
26. 为了对文件系统中的文件进行安全管理，任何一个用户在进入系统时都必须进行注册，这一级安全管理是( )。  
  A. 系统级    B. 目录级                                  C. 用户级    D. 文件级
27. 【2014统考真题】在一个文件被用户进程首次打开的过程中，操作系统需做的是( )。  
  A. 将文件内容读到内存中  
  B. 将文件控制块读到内存中  
  C. 修改文件控制块中的读写权限  
  D. 将文件的数据缓冲区首指针返回给用户进程
28. 【2017统考真题】某文件系统中，针对每个文件，用户类别分为4类：安全管理员、文件主、文件主的伙伴、其他用户；访问权限分为5种：完全控制、执行、修改、读取、写入。若文件控制块中用二进制位串表示文件权限，为表示不同类别用户对一个文件的访问权限，则描述文件权限的位数至少应为( )。

A. 5

B. 9

C. 12

D. 20

29. 下面的说法中，错误的是（ ）。

- I. 一个文件在同一系统中、不同的存储介质上的复制文件，应采用同一种物理结构
  - II. 对一个文件的访问，常由用户访问权限和用户优先级共同限制
  - III. 文件系统采用树形目录结构后，对于不同用户的文件，其文件名应该不同
  - IV. 为防止系统故障造成系统内文件受损，常采用存取控制矩阵方法保护文件
- A. II                    B. I、III                    C. I、III、IV            D. 全选

30. 【2018 统考真题】下列优化方法中，可以提高文件访问速度的是（ ）。

- I. 提前读                    II. 为文件分配连续的簇
  - III. 延迟写                    IV. 采用磁盘高速缓存
- A. 仅 I、II                    B. 仅 II、III                    C. 仅 I、III、IV            D. I、II、III、IV

31. 【2020 统考真题】若多个进程共享同一个文件 F，则下列叙述中，正确的是（ ）。

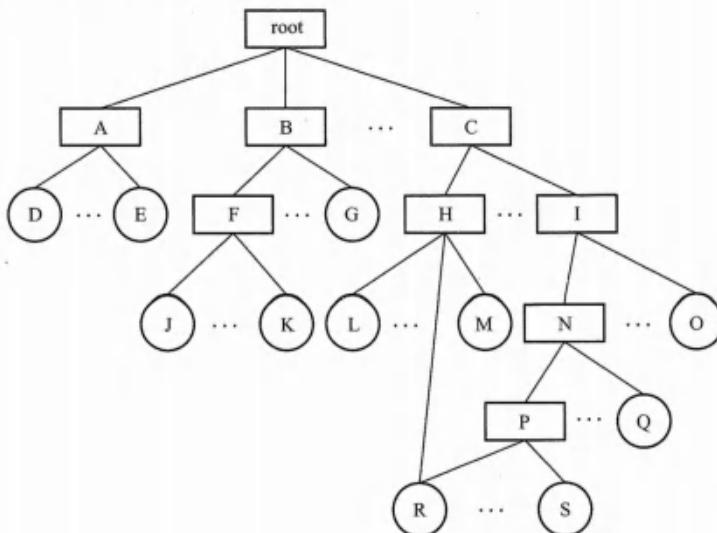
- A. 各进程只能用“读”方式打开文件 F
- B. 在系统打开文件表中仅有一个表项包含 F 的属性
- C. 各进程的用户打开文件表中关于 F 的表项内容相同
- D. 进程关闭 F 时，系统删除 F 在系统打开文件表中的表项

## 二、综合应用题

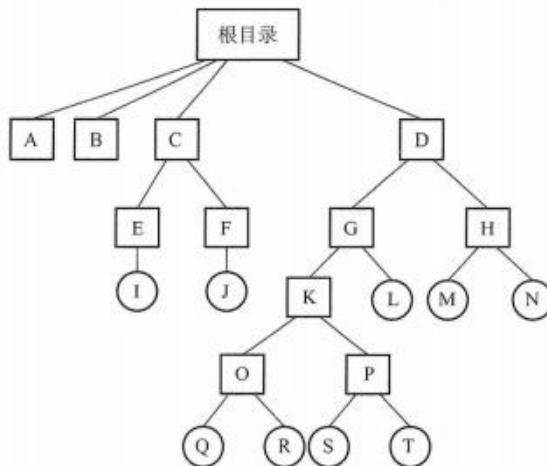
1. 设某文件系统采用两级目录的结构，主目录中有 10 个子目录，每个子目录中有 10 个目录项。在同样多目录的情况下，若采用单级目录结构，所需平均检索目录项数是两级目录结构平均检索目录项数的多少倍？

2. 对文件的目录结构回答以下问题：

- 1) 若一个共享文件可以被用户随意删除或修改，会有什么问题？
  - 2) 若允许用户随意地读、写和修改目录项，会有什么问题？
  - 3) 如何解决上述问题？
3. 有文件系统如下图所示，图中的框表示目录，圆圈表示普通文件。
- 1) 可否建立 F 与 R 的链接？试加以说明。
  - 2) 能否删除 R？为什么？
  - 3) 能否删除 N？为什么？



4. 某树形目录结构的文件系统如下图所示。该图中的方框表示目录，圆圈表示文件。



1) 可否进行下列操作?

- ① 在目录 D 中建立一个文件，取名为 A。
- ② 将目录 C 改名为 A。

2) 若 E 和 G 分别为两个用户的目录:

- ① 用户 E 欲共享文件 Q, 应有什么条件? 如何操作?
- ② 在一段时间内用户 G 主要使用文件 S 和 T. 为简化操作和提高速度, 应如何处理?
- ③ 用户 E 欲对文件 I 加以保护, 不许别人使用, 能否实现? 如何实现?

### 4.1.8 答案与解析

#### 一、单项选择题

1. C

当一个文件系统含有多少级目录时，每访问一个文件，都要使用从树根开始到树叶为止、包括各中间结点名的全路径名。当前目录又称工作目录，进程对各个文件的访问都相对于当前目录进行，而不需要从根目录一层一层地检索，加快了文件的检索速度。选项 A、B 都与相对目录无关；选项 D，文件的读/写速度取决于磁盘的性能。

2. A

为了实现“按名存取”，在文件系统中为每个文件设置用于描述和控制文件的数据结构，称之为文件控制块（FCB）。在文件控制块中，通常包含三类信息，即基本信息、存取控制信息及使用信息。

3. B

从系统角度看，文件系统负责对文件的存储空间进行组织、分配，负责文件的存储并对存入文件进行保护、检索。从用户角度看，文件系统根据一定的格式将用户的文件存放到文件存储器中适当的地方，当用户需要使用文件时，系统根据用户所给的文件名能够从文件存储器中找到所需要的文件。

4. A

一个文件对应一个 FCB，而一个文件目录项就是一个 FCB。

5. A

打开文件操作是将该文件的 FCB 存入内存的活跃文件目录表，而不是将文件内容复制到主

存，找到指定文件目录是打开文件之前的操作。

6. D

UNIX 操作系统中，所有设备都被视为特殊的文件，因为 UNIX 操作系统控制和访问外部设备的方式和访问一个文件的方式是相同的。

7. D

逻辑文件有两种：无结构文件（流式文件）和有结构式文件。连续文件和链接文件都属于文件的物理结构，而系统文件是按文件用途分类的。

8. D

文件结构包括逻辑结构和物理结构。逻辑结构是用户组织数据的结构形式，数据组织形式来自需求，而物理结构是操作系统组织物理存储块的结构形式。

因此说，逻辑文件的组织形式取决于用户，物理结构的选择取决于文件系统设计者针对硬件结构（如磁带介质很难实现链接结构和索引结构）所采取的策略（即 A 和 B 选项）。

9. B

索引文件由逻辑文件和索引表组成。文件的逻辑结构和物理结构都有索引的概念，引入逻辑索引和物理索引的目的是截然不同的。逻辑索引的目的是加快文件数据的定位，是从用户角度出发的，而物理索引的主要目的是管理不连续的物理块，是从系统管理的角度出发的。

10. B

索引文件由逻辑文件和索引表组成，对索引文件存取时，必须先查找索引表。索引项只包含每条记录的长度和在逻辑文件中的起始位置。因为每条记录都要有一个索引项，因此提高了存储代价。

11. C

最好的情况是有  $\sqrt{10000} = 100$  组，每组有 100 条记录，因此顺序查找时平均查找记录个数 =  $50 + 50 = 100$ 。

12. A

对于 I，当所读文件的数据不在内存时，产生中断（缺页中断），原进程进入阻塞态，直到所需数据从外存调入内存后，才将该进程唤醒。对于 II，read 系统调用通过陷入将 CPU 从用户态切换到核心态，从而获取操作系统提供的服务。对于 III，要读一个文件，首先要用 open 系统调用打开该文件。open 中的参数包含文件的路径名与文件名，而 read 只需使用 open 返回的文件描述符，并不使用文件名作为参数。read 要求用户提供三个输入参数：① 文件描述符 fd；② buf 缓冲区首址；③ 传送的字节数 n。read 的功能是试图从 fd 所指示的文件中读入 n 个字节的数据，并将它们送至由指针 buf 所指示的缓冲区中。

13. A

此文件所在目录下可能还存在其他文件，因此删除文件时不能（也不需要）删除文件所在的目录，而与此文件关联的目录项和文件控制块需要随着文件一同删除，同时释放文件关联的内存缓冲区。

14. A

相对路径是从当前目录出发到所找文件通路上所有目录名和数据文件名用分隔符连接起来而形成的，注意与绝对路径的区别。

15. D

目录文件是 FCB 的集合，一个目录中既可能有子目录，又可能有数据文件，因此目录文件中存放的是子目录和数据文件的信息。

16. C

文件目录项即 FCB，通常由文件基本信息、存取控制信息和使用信息组成。基本信息包括文件物理位置。文件目录项显然不包括 FCB 的物理位置信息。

17. C

在文件系统中采用多级目录结构后，符合了多层次管理的需要，提高了文件查找的速度，还允许用户建立同名文件。因此，多级目录结构的采用解决了命名冲突。

18. A

在单级目录文件中，每当新建一个文件时，必须先检索所有的目录项，以保证新文件名在目录中是唯一的。所以单级目录结构无法解决文件重名问题。

19. B

UNIX 采用树形目录结构，文件信息存放在索引结点中。超级块是用来描述文件系统的，具体可参见本章的 4.4 节。

20. A

文件保护是针对文件访问权限的保护。

21. B

建立符号链接时，引用计数值直接复制；建立硬链接时，引用计数值加 1。删除文件时，删除操作对于符号链接是不可见的，这并不影响文件系统，当以后再通过符号链接访问时，发现文件不存在，直接删除符号链接；但对于硬链接则不可直接删除，引用计数值减 1，若值不为 0，则不能删除此文件，因为还有其他硬链接指向此文件。

当建立 F2 时，F1 和 F2 的引用计数值都为 1。当再建立 F3 时，F1 和 F3 的引用计数值就都变成了 2。当后来删除 F1 时，F3 的引用计数值为  $2 - 1 = 1$ ，F2 的引用计数值不变。

22. B

硬链接指通过索引结点进行连接。一个文件在物理存储器上有一个索引结点号。存在多个文件名指向同一个索引结点的情况，II 正确。两个进程各自维护自己的文件描述符，III 正确，I 错误。所以选择 B。

23. D

在文件系统中，口令、存取控制和用户权限表都是常用的文件保护方法。

24. A

对于这道题，只要能区分用户的访问权限和用户优先级，就能得到正确的答案。用户访问权限是指用户有没有权限访问该文件，而用户优先级是指在多个用户同时请求该文件时应该先满足谁。比如，图书馆的用户排队借一本书，某用户可能有更高的优先级，即他排在队伍的前面，但有可能轮到他时被告知他没有借阅那本书的权限。

文件的属性包括保存在 FCB 中对文件访问的控制信息。

25. D

相对于加密保护机制，访问控制机制的安全性较差。因为访问控制的级别和保护力度较小，因此它的灵活性相对较高。若访问控制不由系统实现，则系统本身的安全性就无法保证。加密机制若由系统实现，则加密方法将无法扩展。

26. A

系统级安全管理包括注册和登录。另外，通过“进入系统时”这个关键词也可推测出正确答案。

27. B

一个文件被用户进程首次打开即被执行了 open 操作，会把文件的 FCB 调入内存，而不会把文件内容读到内存中，只有进程希望获取文件内容时才会读入文件内容；C、D 明显错误，选 B。

28. D

可以把用户访问权限抽象为一个矩阵，行代表用户，列代表访问权限。这个矩阵有 4 行 5 列，1 代表 true，0 代表 false，所以需要 20 位，选 D。

29. D

I 错误：一个文件存放在磁带中时，通常采用连续存放方法，文件在硬盘上一般不采用连续存放方法，不同的文件系统存放的方法是不一样的。

II 错误：对一个文件的访问，常由用户访问权限和文件属性共同限制。用户优先级与用户是否有权限访问这个文件没有相关关系。

III 错误：文件系统采用树形目录结构后，对于不同用户的文件，其文件名可以不同，也可以相同。

IV 错误：防止受损常采用备份的方法保护文件，而存取控制矩阵的方法用于多用户之间的存取权限保护。

30. D

II 和 IV 显然均能提高文件访问速度。对于 I，提前读是指在读当前盘块时，将下一个可能要访问的盘块数据读入缓冲区，以便需要时直接从缓冲区中读取，提高了文件的访问速度。对于 III，延迟写是指先将数据写入缓冲区，并置上“延迟写”标志，以备不久之后访问，当缓冲区需要再次被分配出去时，才将缓冲区数据写入磁盘，减少了访问磁盘的次数，提高了文件的访问速度，III 也正确，答案选 D。

31. B

多个进程可以同时以“读”或“写”的方式打开文件，操作系统并不保证写操作的互斥性，进程可通过系统调用对文件加锁，保证互斥写（读者-写者问题），A 错误。整个系统只有一个系统打开文件表，同一个文件打开多次只需改变引用计数，B 正确。用户进程的打开文件表关于同一个文件不一定相同，例如读写指针位置不一定相同，C 错误。进程关闭文件时，文件的引用计数减 1，引用计数变为 0 时才删除系统打开文件表中的表项，D 错误。

## 二、综合应用题

1. 解答：

依题意，文件系统中共有  $10 \times 10 = 100$  个目录，若采用单级目录结构，目录表中有 100 个目录项，在检索一个文件时，平均检索的目录项数 = 目录项/2 = 50。采用两级目录结构时，主目录有 10 个目录项，每个子目录均有 10 个目录项，每级平均检索 5 个目录项，即检索一个文件时平均检索 10 个目录项，所以来采用单级目录结构所需检索目录项数是两级目录结构检索目录项数的  $50/10 = 5$  倍。

2. 解答：

- 1) 将有可能导致共享该文件的其他用户无文件可用，或使用了不想使用的文件，导致发生错误。
- 2) 用户可以通过修改目录项来改变对文件的存取权限，从而非法地使用系统的文件；另外，对目录项不负责任的修改会造成管理上的混乱。
- 3) 解决办法是不允许用户直接执行上述操作，而必须通过系统调用来执行这些操作。

3. 解答：

- 1) 可以建立链接。因为 F 是目录而 R 是文件，所以可以建立 R 到 F 的符号链接。除了符号

链接，也可以通过硬链接的方式

- 2) 不一定能删除 R。由于 R 被多个目录共享，能否删除 R 取决于文件系统实现共享的方法。若采用基于索引结点的共享方法，则因删除后存在指针悬空问题而不能删除 R 结点。若采用基于符号共享的方法，则可以删除 R 结点。
- 3) 不一定能删除 N。由于 N 的子目录中存在共享文件 R，而 R 结点本身不一定能被删除，所以 N 也不一定能被删除。

#### 4. 解答：

- 1) ① 由于目录 D 中没有已命名为 A 的文件，因此在目录 D 中，可以建立一个名为 A 的文件。② 因为在文件系统的根目录下已存在一个名为 A 的目录，所以根目录下的目录 C 不能改名为 A。
- 2) ① 用户 E 欲共享文件 Q，需要用户 E 有访问文件 Q 的权限。在访问权限许可的情况下，用户 E 可通过相应路径来访问文件 Q，即用户 E 通过自己的主目录 E 找到其父目录 C，再访问目录 C 的父目录（根目录），然后依次通过目录 D, G, K 和 O 访问文件 Q。若用户 E 的当前目录为 E，则访问路径为 `../..D/G/K/O/Q`，其中符号“..”表示父目录，符号“/”用于分隔路径中的各目录名。② 用户 G 需要通过依次访问目录 K 和目录 P，才能访问文件 S 和文件 T。为了提高文件访问速度，可在目录 G 下建立两个链接文件，分别链接到文件 S 和文件 T 上。这样用户 G 就可直接访问这两个文件。③ 用户 E 可以修改文件 I 的存取控制表来对文件 I 加以保护，不让别的用户使用。具体实现方法是：在文件 I 的存取控制表中，只留下用户 E 的访问权限，其他用户对该文件无操作权限，从而达到不让其他用户访问的目的。

## 4.2 文件系统实现

在学习本节时，请读者思考以下问题：

- 1) 在目录中查找某个文件可以使用什么方法？
- 2) 文件的逻辑结构和物理结构有何区别？单个文件的逻辑结构和物理结构之间是否存在某些制约关系？

上节介绍了目录和文件的逻辑结构，本节将介绍文件物理结构和目录的实现。建议读者阅读之前先回顾上节的内容，并自己思考相应功能的实现方法，在学习过程中和本节的方法进行对比，这样能更好地理解本节的内容。

### 4.2.1 文件系统层次结构

现代操作系统有多种文件系统类型（如 FAT32, NTFS, ext2, ext3, ext4 等），因此文件系统的层次结构也不尽相同。图 4.9 是一种合理的层次结构。

#### 1. 用户调用接口

文件系统为用户提供与文件及目录有关的调用，如新建、打开、读写、关闭、删除文件，建立、删除目录等。此层由若干程序模块组成，每个模块对应一条系统调用，用户发出系统



图 4.9 文件系统层次结构

调用时，控制即转入相应的模块。

## 2. 文件目录系统

文件目录系统的主要功能是管理文件目录，其任务有管理活跃文件目录表、管理读写状态信息表、管理用户进程的打开文件表、管理与组织存储设备上的文件目录结构、调用下一级存取控制模块。

## 3. 存取控制验证模块

实现文件保护主要由该级软件完成，它把用户的访问要求与 FCB 中指示的访问控制权限进行比较，以确认访问的合法性。

## 4. 逻辑文件系统与文件信息缓冲区

逻辑文件系统与文件信息缓冲区的主要功能是，根据文件的逻辑结构将用户要读写的逻辑记录转换成文件逻辑结构内的相应块号。

## 5. 物理文件系统

物理文件系统的主要功能是把逻辑记录所在的相对块号转换成实际的物理地址。

## 6. 辅助分配模块

分配模块的主要功能是管理辅存空间，即负责分配辅存空闲空间和回收辅存空间。

## 7. 设备管理程序模块

设备管理程序模块的主要功能是分配设备、分配设备读写用缓冲区、磁盘调度、启动设备、处理设备中断、释放设备读写缓冲区、释放设备等。

对于文件管理系统的层次结构我们不能忽略，因为它是重要考点之一，当然也不需要死记硬背，我们可以通过用户请求访问某个文件时发生的一系列事情来辅助记忆文件系统的层次结构。

例如，用户要查看文件 F 中的内容，对操作系统发出命令（操作系统有面向用户的接口），于是就经过了第 0 级的用户调用接口。

操作系统得到命令后，需要查找目录以查找文件 F 的索引信息，可能是 FCB，也可能是索引结点，经过了第 1 级文件目录系统。

通过目录找到文件 FCB 后，需要查看文件 FCB 上的信息，看看那个用户有没有访问该文件的权限，于是经过了存取控制验证模块。

用户通过验证后，就真正开始寻址。经历第 3 章的学习后，我们有这样的意识：操作系统的寻址往往要先得到逻辑地址，再得到物理地址，于是在开始寻址时，操作系统经过逻辑文件系统与文件信息缓冲区，得到了相应文件的内容的逻辑地址。

把逻辑地址转换为物理地址，是在物理文件系统中完成的。

至此为止，寻址就已完成。寻址完成后，我们关心的是找到的这块空间应该如何管理，若要释放这块空间，则任务就交给辅助分配模块，若要把这块空间分配给设备用于输入/输出，则把任务交给设备管理程序模块。

### 4.2.2 目录实现

在读文件前，必须先打开文件。打开文件时，操作系统利用路径名找到相应目录项，目录项中提供了查找文件磁盘块所需要的信息。目录实现的基本方法有线性列表和哈希表两种，要注意目录的实现就是为了查找，因此线性列表实现对应线性查找，哈希表的实现对应散列查找。

#### 1. 线性列表

最简单的目录实现方法是使用存储文件名和数据块指针的线性表。创建新文件时，必须首先

搜索目录表以确定没有同名的文件存在，然后在目录表后增加一个目录项。删除文件则根据给定的文件名搜索目录表，接着释放分配给它的空间。重用目录项有许多方法：可以将目录项标记为不再使用，或将它加到空闲目录项表上，还可以将目录表中的最后一个目录项复制到空闲位置，并降低目录表长度。采用链表结构可以减少删除文件的时间，其优点在于实现简单，不过由于线性表的特殊性，比较费时。

## 2. 哈希表

哈希表根据文件名得到一个值，并返回一个指向线性列表中元素的指针。这种方法的优点是查找非常迅速，插入和删除也较简单，不过需要一些预备措施来避免冲突。最大的困难是哈希表长度固定以及哈希函数对表长的依赖性。

目录查询是通过在磁盘上反复搜索完成的，需要不断地进行 I/O 操作，开销较大。所以如前所述，为了减少 I/O 操作，把当前使用的文件目录复制到内存，以后要使用该文件时只需在内存中操作，因此降低了磁盘操作次数，提高了系统速度。

### 4.2.3 文件实现——文件分配方式

前面说过，文件实际上是一种抽象数据类型，我们要研究它的逻辑结构、物理结构，以及关于它的一系列操作（不是统考关注的内容）。文件的实现就是研究文件的物理结构，即文件数据在物理存储设备上是如何分布和组织的。同一个问题有两个方面的回答：一是文件的分配方式，讲的是对磁盘非空闲块的管理；二是文件存储空间管理，讲的是对磁盘空闲块的管理。

文件分配对应于文件的物理结构，是指如何为文件分配磁盘块。常用的磁盘空间分配方法有三种：连续分配、链接分配和索引分配。有的系统（如 RDOS 操作系统）对三种方法都支持，但更普遍的是一个系统只支持一种方法。对于本节的内容，读者要注意与文件的逻辑结构区分，从历年经验来看，这是很多读者容易搞混的地方（读者复习完数据结构后，应该了解线性表、顺序表和链表之间的关系，类比到这里就不易混淆）。

#### 1. 连续分配

连续分配方法要求每个文件在磁盘上占有一组连续的块，如图 4.10 所示。磁盘地址定义了磁盘上的一个线性排序。这种排序使作业访问磁盘时需要的寻道数和寻道时间最小。

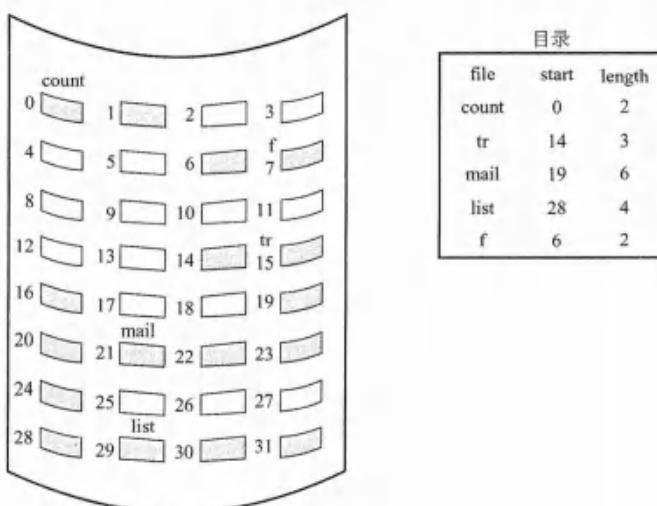


图 4.10 连续分配

文件的连续分配可以用第一块的磁盘地址和连续块的数量来定义。若文件长  $n$  块并从位置  $b$  开始，则该文件将占有块  $b, b+1, b+2, \dots, b+n-1$ 。一个文件的目录条目包括开始块的地址和该文件所分配区域的长度。

连续分配支持顺序访问和直接访问。其优点是实现简单、存取速度快。缺点是文件长度不宜动态增加，因为一个文件末尾后的盘块可能已分配给其他文件，一旦需要增加，就需要大量移动盘块。此外，反复增删文件后会产生外部碎片（与内存管理分配方式中的碎片相似），且很难确定一个文件需要的空间大小，因而只适用于长度固定的文件。

## 2. 链接分配

链接分配采取离散分配的方式，消除了外部碎片，因此显著提高了磁盘空间的利用率；又因为根据文件的当前需求为其分配必需的盘块，当文件动态增长时，可以动态地再为它分配盘块，因此无须事先知道文件的大小。此外，对文件的增、删、改也非常方便。链接分配又可以分为隐式链接和显式链接两种形式。

隐式链接如图 4.11 所示。每个文件对应一个磁盘块的链表；磁盘块分布在磁盘的任何地方，除最后一个盘块外，每个盘块都有指向下一个盘块的指针，这些指针对用户是透明的。目录包括文件第一块的指针和最后一块的指针。

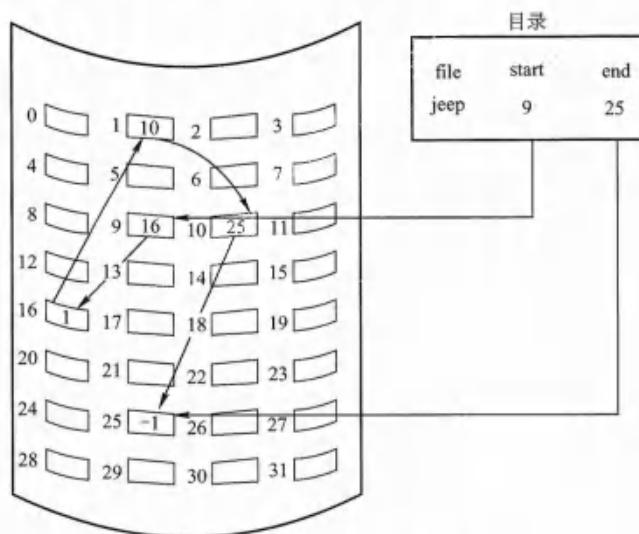


图 4.11 隐式链接分配

创建新文件时，目录中增加一个新条目。每个目录项都有一个指向文件首块的指针。该指针初始化为 NULL 以表示空文件，大小字段为 0。写文件会通过空闲空间管理系统找到空闲块，将该块链接到文件的尾部，以便写入。读文件则通过块到块的指针顺序读块。

隐式链接分配的缺点是无法直接访问盘块，只能通过指针顺序访问文件，且盘块指针会消耗一定的存储空间。隐式链接分配的稳定性也是一个问题，系统在运行过程中由于软件或硬件错误导致链表中的指针丢失或损坏，会导致文件数据的丢失。

显式链接是指把用于链接文件各物理块的指针，从每个物理块的块末尾中提取出来，显式地存放在内存的一张链接表中。该表在整个磁盘中仅设置一张，称为文件分配表（File Allocation Table, FAT）。每个表项中存放对应块的下一块链接指针，即下一个盘块号。文件的第一个盘块号记录在目录中，后续的盘块可通过查 FAT 找到。例如，某磁盘共有 100 个磁盘块，存放了两个文

件：文件“aaa”占三个盘块，依次是2→8→5；文件“bbb”占两个盘块，依次是7→1。其余盘块都是空闲盘块，则该磁盘的FAT表如图4.12所示。

不难看出，FAT的表项与全部磁盘块一一对应，并且可以用一个特殊的数字-1表示文件的最后一块，用-2表示这个磁盘块是空闲的（当然也可指定为-3，-4）。因此，文件分配表（FAT）不仅记录了文件各块之间的先后链接关系，同时还标记了空闲的磁盘块，操作系统也可以通过FAT对文件存储空间进行管理。当某进程请求操作系统分配一个磁盘块时，操作系统只需从FAT中找到-2的表项，并将对应的磁盘块分配给进程即可。

FAT表在系统启动时就会被读入内存，因此查找FAT的过程是在内存中进行的，因此不仅显著地提高了检索速度，而且明显减少了访问磁盘的次数。

### 3. 索引分配

链接分配解决了连续分配的外部碎片和文件大小管理的问题。但是，链接分配不能有效支持直接访问（FAT除外）。索引分配解决了这个问题，它把每个文件的所有盘块号都集中放在一起构成索引块（表），如图4.13所示。

文件名	.....	起始块号
aaa	...	2
bbb	...	7

盘块号	下一块
0	-2
1	-1
2	8
3	-2
4	-2
5	-1
6	-2
7	1
8	5
9	-2
...	...
98	-2
99	-2

图4.12 文件分配表

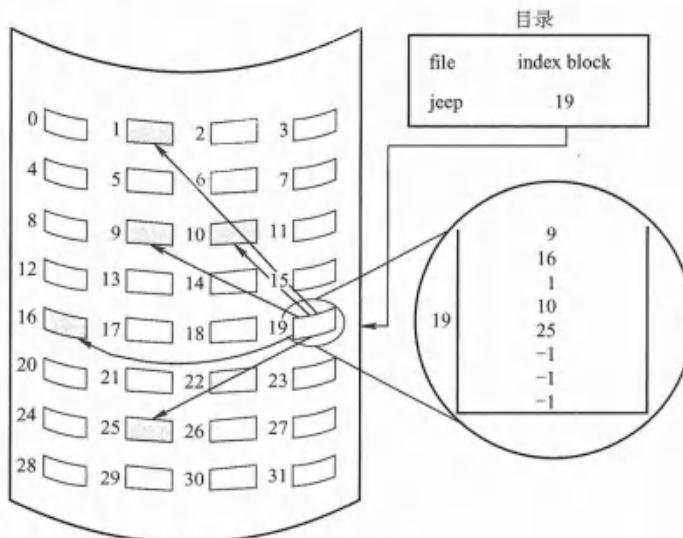


图4.13 索引分配

每个文件都有其索引块，这是一个磁盘块地址的数组。索引块的第*i*个条目指向文件的第*i*块。目录条目包括索引块的地址。要读第*i*块，通过索引块的第*i*个条目的指针来查找和读入所需的块。

创建文件时，索引块的所有指针都设为空。首次写入第*i*块时，先从空闲空间中取得一个块，再将其地址写到索引块的第*i*个条目。索引分配支持直接访问，且没有外部碎片问题。其缺点是由于索引块的分配，增加了系统存储空间的开销。索引块的大小是一个重要的问题，每个文件必须有一个索引块，因此索引块应尽可能小，但索引块太小就无法支持大文件。可以采用以下机制

来处理这个问题。

- **链接方案。**一个索引块通常为一个磁盘块，因此它本身能直接读写。为了处理大文件，可以将多个索引块链接起来。
- **多层索引。**多层索引使第一层索引块指向第二层的索引块，第二层索引块再指向文件块。这种方法根据最大文件大小的要求，可以继续到第三层或第四层。例如，4096B 的块，能在索引块中存入 1024 个 4B 的指针。两层索引允许 1048576 个数据块，即允许最大文件为 4GB。
- **混合索引。**将多种索引分配方式相结合的分配方式。例如，系统既采用直接地址，又采用单级索引分配方式或两级索引分配方式（混合索引是本章最综合的高频考点，可先学习本章疑难点 4，然后回来接着学习）。

表 4.2 是三种分配方式的比较。

表 4.2 文件三种分配方式的比较

	访问第 $n$ 条记录	优 点	缺 点
连续分配	需访问磁盘 1 次	顺序存取时速度快，文件定长时可根据文件起始地址及记录长度进行随机访问	文件存储要求连续的存储空间，会产生碎片，不利于文件的动态扩充
链接分配	需访问磁盘 $n$ 次	可解决外存的碎片问题，提高外存空间的利用率，动态增长较方便	只能按照文件的指针链顺序访问，查找效率低，指针信息存放消耗外存空间
索引分配	$m$ 级需访问磁盘 $m+1$ 次	可以随机访问，文件易于增删	索引表增加存储空间的开销，索引表的查找策略对文件系统效率影响较大

此外，访问文件需要两次访问外存——首先要读取索引块的内容，然后访问具体的磁盘块，因而降低了文件的存取速度。为了解决这一问题，通常将文件的索引块读入内存的缓冲区中，以加快文件的访问速度。

### 4.2.3 文件实现——文件存储空间管理

#### (1) 文件存储器空间的划分与初始化

一般来说，一个文件存储在一个文件卷中。文件卷可以是物理盘的一部分，也可以是整个物理盘，支持超大型文件的文件卷也可由多个物理盘组成，如图 4.14 所示。

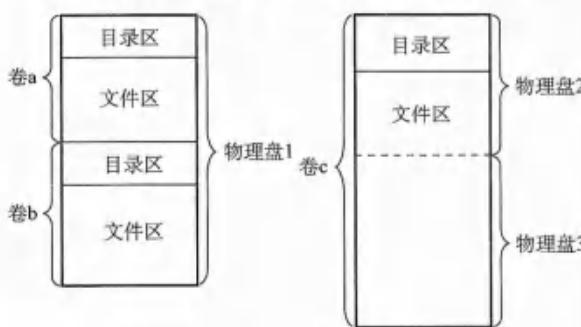


图 4.14 逻辑卷与物理盘的关系

建立空闲空间管理表格及存放逻辑卷信息的超级块。

#### (2) 文件存储器空间管理

文件存储设备分成许多大小相同的物理块，并以块为单位交换信息，因此，文件存储设备的管理实质上是对空闲块的组织和管理，它包括空闲块的组织、分配与回收等问题。

### 1. 空闲表法

空闲表法属于连续分配方式，它与内存的动态分配方式类似，为每个文件分配一块连续的存储空间。系统为外存上的所有空闲区建立一张空闲盘块表，每个空闲区对应于一个空闲表项，其中包括表项序号、该空闲区第一个盘块号、该区的空闲盘块数等信息。再将所有空闲区按其起始盘块号递增的次序排列，如表 4.3 所示。

空闲盘区的分配与内存的动态分配类似，同样采用首次适应算法、循环首次适应算法等。例如，在系统为某新创建的文件分配空闲盘块时，先顺序地检索空闲盘块表的各表项，直至找到第一个其大小能满足要求的空闲区，再将该盘区分配给用户，同时修改空闲盘块表。

系统在对用户所释放的存储空间进行回收时，也采取类似于内存回收的方法，即要考虑回收区是否与空闲表中插入点的前区和后区相邻接，对相邻接者应予以合并。

### 2. 空闲链表法

将所有空闲盘区拉成一条空闲链，根据构成链所用的基本元素不同，可把链表分成两种形式：空闲盘块链和空闲盘区链。

空闲盘块链将磁盘上的所有空闲空间以盘块为单位拉成一条链。当用户因创建文件而请求分配存储空间时，系统从链首开始，依次摘下适当数目的空闲盘块分配给用户。当用户因删除文件而释放存储空间时，系统将回收的盘块依次插入空闲盘块链的末尾。这种方法的优点是分配和回收一个盘块的过程非常简单，但在为一个文件分配盘块时可能要重复多次操作。

空闲盘区链将磁盘上的所有空闲盘区（每个盘区可包含若干盘块）拉成一条链。在每个盘区上除含有用于指示下一个空闲盘区的指针外，还应有能指明本盘区大小（盘块数）的信息。分配盘区的方法与内存的动态分区分配类似，通常采用首次适应算法。在回收盘区时，同样也要将回收区与相邻接的空闲盘区合并。

### 3. 位示图法

位示图利用二进制的一位来表示磁盘中一个盘块的使用情况，磁盘上所有的盘块都有一个二进制位与之对应。当其值为“0”时，表示对应的盘块空闲；当其值为“1”时，表示对应的盘块已分配。位示图法示意如图 4.15 所示。

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	1	1	0	0	0	1	1	1	0	0	1	0	0	1	1	0
2	0	0	0	1	1	1	1	1	1	0	0	0	0	1	1	1
3	1	1	1	0	0	0	1	1	1	1	1	1	0	0	0	0
4																
⋮																
16																

图 4.15 位示图法示意图

盘块的分配：

- ① 顺序扫描位示图，从中找出一个或一组其值为“0”的二进制位。
- ② 将找到的一个或一组二进制位，转换成与之对应的盘块号。若找到的其值为“0”的二进制位位于位示图的第  $i$  行、第  $j$  列，则其相应的盘块号应按下式计算 ( $n$  代表每行的位数)：

$$b = n(i-1) + j$$

③ 修改位示图，令  $\text{map}[i, j] = 1$ 。

盘块的回收：

① 将回收盘块的盘块号转换成位示图中的行号和列号。转换公式为

$$i = (b - 1) \text{ DIV } n + 1$$

$$j = (b - 1) \text{ MOD } n + 1$$

② 修改位示图，令  $\text{map}[i, j] = 0$ 。

#### 4. 成组链接法

空闲表法和空闲链表法都不适用于大型文件系统，因为这会使空闲表或空闲链表太大。在 UNIX 系统中采用的是成组链接法，这种方法结合了空闲表和空闲链表两种方法，克服了表太大的缺点。其大致思想是：把顺序的  $n$  个空闲扇区地址保存在第一个空闲扇区内，其后一个空闲扇区内则保存另一顺序空闲扇区的地址，如此继续，直至所有空闲扇区均予以链接。系统只需要保存一个指向第一个空闲扇区的指针。假设磁盘最初全为空闲扇区，其成组链接如图 4.16 所示。通过这种方式可以迅速找到大批空闲块地址。

表示文件存储器空闲空间的“位向量”表或第一个成组链块，以及卷中的目录区、文件区划分信息都需要存放在辅存储器中，一般放在卷头位置，在 UNIX 系统中称为超级块。在对卷中的文件进行操作前，超级块需要预先读入系统空闲的主存，并且经常保持主存超级块与辅存卷中超级块的一致性。

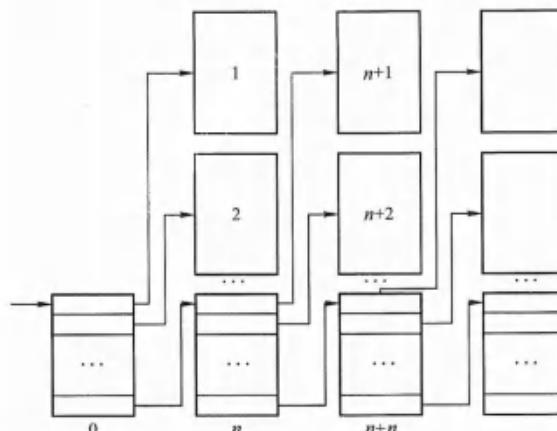


图 4.16 成组链接法示意图

注意：本书如无特别提示，所使用的位示图法中行和列都从 1 开始编号。特别注意，若题目中指明从 0 开始编号，则上述计算方法要进行相应调整。

#### 4.2.5 本节小结

本节开头提出的问题的参考答案如下。

1) 在目录中查找某个文件可以使用什么方法？

可以采用线性列表法或哈希表法。线性列表把文件名组织成一个线性表，查找时依次与线性表中的每个表项进行比较。若把文件名按序排列，则使用折半查找法可以降低平均的查找时间，但建立新文件时会增加维护线性表的开销。哈希表用文件名通过哈希函数得到一个指向文件的指针，这种方法非常迅速，但要注意避免冲突。

2) 文件的逻辑结构和物理结构有何区别? 单个文件的逻辑结构和物理结构之间是否存在着某些制约关系?

文件的逻辑结构是用户可见的结构, 即用户使用文件的结构。文件的物理结构是文件在存储器上的组织结构, 它表示一个文件在辅存上安置、链接、编目的方法。它和文件的存取方法以及辅存设备的特性等都有着密切的联系。单个文件的逻辑结构和物理结构之间虽无明显的制约或关联关系, 但是如果物理结构选择不慎, 也很难体现出逻辑结构的特点, 比如一个逻辑结构是顺序结构, 而物理结构是隐式链接结构的文件, 即使理论上可以很快找出某条记录的地址, 而实际找时仍然需要在磁盘上一块一块地找。

### 4.2.6 本节习题精选

#### 一、单项选择题

1. 【2009 统考真题】下列文件物理结构中, 适合随机访问且易于文件扩展的是( )。
 

A. 连续结构	B. 索引结构
C. 链式结构且磁盘块定长	D. 链式结构且磁盘块变长
2. 【2010 统考真题】设文件索引结点中有 7 个地址项, 其中 4 个地址项是直接地址索引, 2 个地址项是一级间接地址索引, 1 个地址项是二级间接地址索引, 每个地址项大小为 4B, 若磁盘索引块和磁盘数据块大小均为 256B, 则可表示的单个文件最大长度是( )。
 

A. 33KB	B. 519KB	C. 1057KB	D. 16516KB
---------	----------	-----------	------------
3. 以下不适合直接存取的外存分配方式是( )。
 

A. 连续分配	B. 链接分配
C. 索引分配	D. 以上答案都适合
4. 在以下文件的物理结构中, 不利于文件长度动态增长的是( )。
 

A. 连续结构	B. 链接结构	C. 索引结构	D. 散列结构
---------	---------	---------	---------
5. 【2013 统考真题】为支持 CD-ROM 中视频文件的快速随机播放, 播放性能最好的文件数据块组织方式是( )。
 

A. 连续结构	B. 链式结构
C. 直接索引结构	D. 多级索引结构
6. 文件系统中若文件的物理结构采用连续结构, 则 FCB 中有关文件的物理位置的信息应包括( )。
 

I. 首块地址	II. 文件长度	III. 索引表地址	
A. 仅 I	B. I、II	C. II、III	D. I、III
7. 在磁盘上, 最容易导致存储碎片发生的物理文件结构是( )。
 

A. 隐式链接	B. 顺序存放
C. 索引存放	D. 显式链接
8. 有些操作系统中将文件描述信息从目录项中分离出来, 这样做的好处是( )。
 

A. 减少读文件时的 I/O 信息量	B. 减少写文件时的 I/O 信息量
C. 减少查找文件时的 I/O 信息量	D. 减少复制文件时的 I/O 信息量
9. 位示图可用于( )。
 

A. 文件目录的查找	B. 磁盘空间的管理
C. 主存空间的管理	D. 文件的保密
10. 文件系统采用两级索引分配方式。若每个磁盘块的大小为 1KB, 每个盘块号占 4B, 则该系统中, 单个文件的最大长度是( )。
 

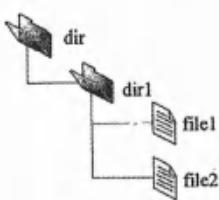
A. 64MB	B. 128MB
---------	----------

- C. 32MB                          D. 以上答案都不对
11. 【2013 统考真题】若某文件系统索引结点 (inode) 中有直接地址项和间接地址项，则下列选项中，与单个文件长度无关的因素是 ( )。
- A. 索引结点的总数              B. 间接地址索引的级数  
C. 地址项的个数              D. 文件块大小
12. 一个文件系统中，其 FCB 占 64B，一个盘块大小为 1KB，采用一级目录。假定文件目录中有 3200 个目录项，则查找一个文件平均需要 ( ) 次访问磁盘。
- A. 50                          B. 54                          C. 100                          D. 200
13. 下列关于目录检索的论述中，正确的是 ( )。
- A. 由于散列法具有较快的检索速度，因此现代操作系统中都用它来替代传统的顺序检索方法  
B. 在利用顺序检索法时，对树形目录应采用文件的路径名，且应从根目录开始逐级检索  
C. 在利用顺序检索法时，只要路径名的一个分量名未找到，就应停止查找  
D. 利用顺序检索法查找完成后，即可得到文件的物理地址
14. 文件的存储空间管理实质上是对 ( ) 的组织和管理。
- A. 文件目录              B. 外存已占用区域              C. 外存空闲区              D. 文件控制块
15. 若用 8 个字 (字长 32 位) 组成的位示图管理内存，假定用户归还一个块号为 100 的内存块时，它对应位示图的位置为 ( )。
- A. 字号为 3，位号为 5              B. 字号为 4，位号为 4  
C. 字号为 3，位号为 4              D. 字号为 4，位号为 5
16. 设有一个记录文件，采用链接分配方式，逻辑记录的固定长度为 100B，在磁盘上存储时采用记录成组分解技术。盘块长度为 512B。若该文件的目录项已经读入内存，则对第 22 个逻辑记录完成修改后，共启动了磁盘 ( ) 次。
- A. 3                          B. 4                          C. 5                          D. 6
17. 物理文件的组织方式是由 ( ) 确定的。
- A. 应用程序              B. 主存容量              C. 外存容量              D. 操作系统
18. 文件系统为每个文件创建一张 ( )，存放文件数据块的磁盘存放位置。
- A. 打开文件表              B. 位图                      C. 索引表                      D. 空闲盘块链表
19. 下面关于索引文件的论述中，正确的是 ( )。
- A. 索引文件中，索引表的每个表项中含有相应记录的关键字和存放该记录的物理地址  
B. 顺序文件进行检索时，首先从 FCB 中读出文件的第一个盘块号；而对索引文件进行检索时，应先从 FCB 中读出文件索引块的开始地址  
C. 对于一个具有三级索引的文件，存取一条记录通常要访问三次磁盘  
D. 文件较大时，无论是进行顺序存取还是进行随机存取，通常索引文件方式都最快
20. 【2015 统考真题】在文件的索引结点中存放直接索引指针 10 个，一级和二级索引指针各 1 个。磁盘块大小为 1KB，每个索引指针占 4B。若某文件的索引结点已在内存中，则把该文件偏移量 (按字节编址) 为 1234 和 307400 处所在的磁盘块读入内存，需访问的磁盘块个数分别是 ( )。
- A. 1, 2                      B. 1, 3                      C. 2, 3                      D. 2, 4
21. 【2015 统考真题】文件系统用位图法表示磁盘空间的分配情况，位图存于磁盘的 32~127 号块中，每个盘块占 1024B，盘块和块内字节均从 0 开始编号。假设要释放的盘块号为

- 409612，则位图中要修改的位所在的盘块号和块内字节序号分别是（ ）。
- A. 81, 1      B. 81, 2      C. 82, 1      D. 82, 2
22. 【2019 统考真题】下列选项中，可用于文件系统管理空闲磁盘块的数据结构是（ ）。
- I. 位图    II. 索引结点    III. 空闲磁盘块链    IV. 文件分配表 (FAT)
- A. 仅 I、II      B. 仅 I、III、IV      C. 仅 I、III      D. 仅 II、III、IV
23. 【2020 统考真题】下列选项中，支持文件长度可变、随机访问的磁盘存储空间分配方式是（ ）。
- A. 索引分配      B. 链接分配      C. 连续分配      D. 动态分区分配
24. 【2020 统考真题】某文件系统的目录项由文件名和索引结点号构成。若每个目录项长度为 64 字节，其中 4 字节存放索引结点号，60 字节存放文件名。文件名由小写英文字母构成，则该文件系统能创建的文件数量的上限为（ ）。
- A.  $2^{26}$       B.  $2^{32}$       C.  $2^{60}$       D.  $2^{64}$

## 二、综合应用题

- 简述文件的外存分配中，连续分配、链接分配和索引分配各自的主要优缺点。
- 在实现文件系统时，为加快文件目录的检索速度，可利用“FCB 分解法”。假设目录文件存放在磁盘上，每个盘块 512B。FCB 占 64B，其中文件名占 8B。通常将 FCB 分解成两部分，第一部分占 10B（包括文件名和文件内部号），第二部分占 56B（包括文件内部号和文件的其他描述信息）。
  - 假设某一目录文件共有 254 个 FCB，试分别给出采用分解法前和分解法后，查找该目录文件的某个 FCB 的平均访问磁盘次数。
  - 一般地，若目录文件分解前占用  $n$  个盘块，分解后改用  $m$  个盘块存放文件名和文件内部号，请给出访问磁盘次数减少的条件。
- 设某文件为链接文件，由 5 个逻辑记录组成，每个逻辑记录的大小与磁盘块的大小相等，均为 512B，并依次存放在 50, 121, 75, 80, 63 号磁盘块上。若要存取文件的第 1569 逻辑字节处的信息，要访问哪个磁盘块？
- 【2011 统考真题】某文件系统为一级目录结构，文件的数据一次性写入磁盘，已写入的文件不可修改，但可多次创建新文件。请回答如下问题。
  - 在连续、链式、索引三种文件的数据块组织方式中，哪种更合适？说明理由。为定位文件数据块，需要在 FCB 中设计哪些相关描述字段？
  - 为快速找到文件，对于 FCB，是集中存储好，还是与对应的文件数据块连续存储好？说明理由。
- 假定磁盘块的大小为 1KB，对于 540MB 的硬盘，其文件分配表 (FAT) 最少需要占用多少存储空间？
- 【2016 统考真题】某磁盘文件系统使用链接分配方式组织文件，簇大小为 4KB。目录文件的每个目录项包括文件名和文件的第一个簇号，其他簇号存放在文件分配表 FAT 中。
  - 假定目录树如下图所示，各文件占用的簇号及顺序如下表所示，其中 dir, dir1 是目录，file1, file2 是用户文件。请给出所有目录文件的内容。
  - 若 FAT 的每个表项仅存放簇号，占 2B，则 FAT 的最大长度为多少字节？该文件系统支持的文件长度最大是多少？
  - 系统通过目录文件和 FAT 实现对文件的按名存取，说明 file1 的 106, 108 两个簇号分别存放在 FAT 的哪个表项中。



文件名	簇号
dir	1
dir1	48
file1	100,106,108
file2	200,201,202

- 4) 假设仅 FAT 和 dir 目录文件已读入内存，若需将文件 dir/dir1/file1 的第 5000 个字节读入内存，则要访问哪几个簇？
7. 有一个文件系统如图 A 所示。图中的方框表示目录，圆圈表示普通文件。根目录常驻内存，目录文件组织成链接文件，不设 FCB，普通文件组织成索引文件。目录表指示下一级文件名及其磁盘地址（各占 2B，共 4B）。下级文件是目录文件时，指示其第一个磁盘块地址。下级文件是普通文件时，指示其 FCB 的磁盘地址。每个目录的文件磁盘块的最后 4B 供拉链使用。下级文件在上级目录文件中的次序在图中为从左至右。每个磁盘块有 512B，与普通文件的一页等长。

普通文件的 FCB 组织如图 B 所示。其中，每个磁盘地址占 2B，前 10 个地址直接指示该文件前 10 页的地址。第 11 个地址指示一级索引表地址，一级索引表中的每个磁盘地址指示一个文件页地址；第 12 个地址指示二级索引表地址，二级索引表中的每个地址指示一个一级索引表地址；第 13 个地址指示三级索引表地址，三级索引表中的每个地址指示一个二级索引表地址。请问：

- 1) 一个普通文件最多可有多少个文件页？
- 2) 若要读文件 J 中的某一页，最多启动磁盘多少次？
- 3) 若要读文件 W 中的某一页，最少启动磁盘多少次？
- 4) 根据 3)，为最大限度地减少启动磁盘的次数，可采用什么方法？此时，磁盘最多启动多少次？

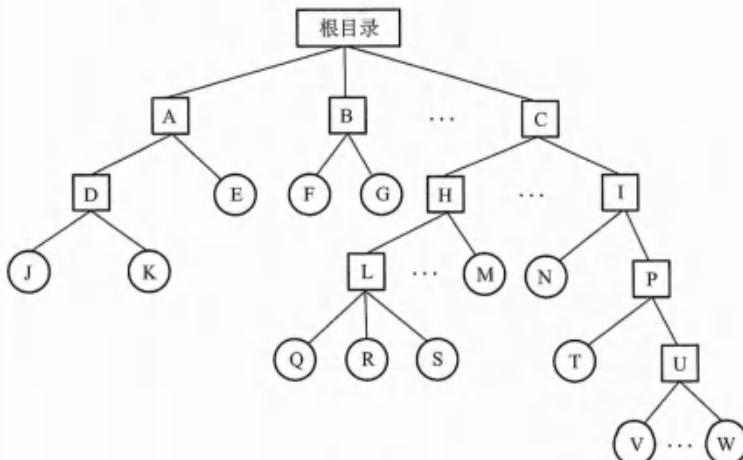


图 A 某树形结构文件系统框图

该文件的有关描述信息	
1	磁盘地址
2	磁盘地址
3	磁盘地址
:	...
11	磁盘地址
12	磁盘地址
13	磁盘地址

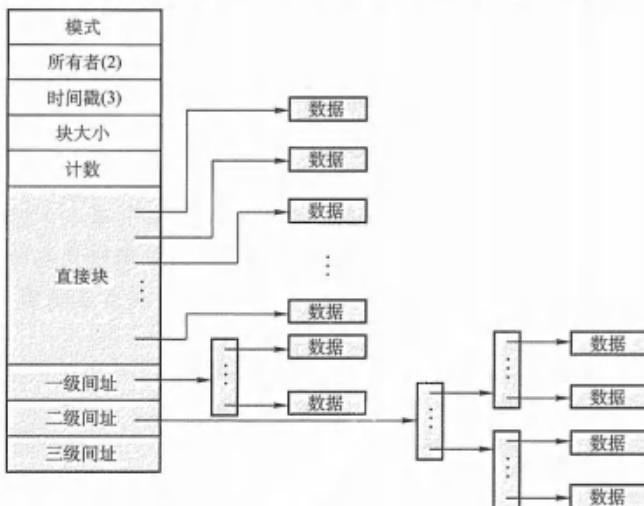
图 B FCB 组织

8. 在 UNIX 操作系统中，给文件分配外存空间采用的是混合索引分配方式，如下图所示。UNIX 系统中的某个文件的索引结点指示出了为该文件分配的外存的物理块的寻找方法。在该索引结点中，有 10 个直接块（每个直接块都直接指向一个数据块），有 1 个一级间

接块、1个二级间接块及1个三级间接块，间接块指向的是一个索引块，每个索引块和数据块的大小均为4KB，而UNIX系统中地址所占空间为4B（指针大小为4B），假设以下问题都建立在该索引结点已在内存中的前提下。

现请回答：

- 1) 文件的大小为多大时可以只用到索引结点的直接块？
- 2) 该索引结点能访问到的地址空间大小总共为多大（小数点后保留2位）？
- 3) 若要读取一个文件的第10000B的内容，需要访问磁盘多少次？
- 4) 若要读取一个文件的第10MB的内容，需要访问磁盘多少次？

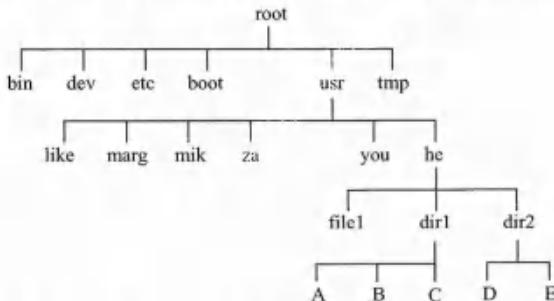


9. 某文件系统采用多级索引的方式组织文件的数据存放，假定在文件的 `i_node` 中设有 13 个地址项，其中直接索引 10 项，一次间接索引项 1 项，二次间接索引项 1 项，三次间接索引项 1 项。数据块的大小为 4KB，磁盘地址用 4B 表示，试问：
  - 1) 这个文件系统允许的最大文件长度是多少？
  - 2) 一个 2GB 大小的文件，在这个文件系统中实际占用多少空间？
10. 一计算机系统利用位示图来管理磁盘文件空间。假定该磁盘组共有 100 个柱面，每个柱面有 20 个磁道，每个磁道分成 8 个盘块（扇区），每个盘块 1KB，位示图如下图所示。

$i,j$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
2	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	
3	1	1	1	1	1	1	0	1	1	1	1	1	1	0	0	
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
...																

- 1) 试给出位示图中位置  $(i, j)$  与对应盘块所在的物理位置（柱面号，磁头号，扇区号）之间的计算公式。假定柱面号、磁头号、扇区号都从 0 开始编号。
- 2) 试说明分配和回收一个盘块的过程。
11. 假定一个盘组共有 100 个柱面，每个柱面上有 16 个磁道，每个磁道分成 4 个扇区。
  - 1) 整个磁盘空间共有多少个存储块？
  - 2) 若用字长 32 位的单元来构造位示图，共需要多少个字？

- 3) 位示图中第 18 个字的第 16 位对应的块号是多少?
12. 文件采用多重索引结构搜索文件内容。设块长为 512B, 每个块号长 2B, 若不考虑逻辑块号在物理块中所占的位置, 分别计算二级索引和三级索引时可寻址的文件最大长度。
13. 【2012 统考真题】某文件系统空间的最大容量为 4TB ( $1TB = 2^{40}B$ ), 以磁盘块为基本分配单位。磁盘块大小为 1KB。文件控制块 (FCB) 包含一个 512B 的索引表区。请回答下列问题:
- 1) 假设索引表区仅采用直接索引结构, 索引表区存放文件占用的磁盘块号, 索引表项中块号最少占多少字节? 可支持的单个文件的最大长度是多少字节?
  - 2) 假设索引表区采用如下结构: 第 0~7 字节采用<起始块号, 块数>格式表示文件创建时预分配的连续存储空间。其中起始块号占 6B, 块数占 2B, 剩余 504B 采用直接索引结构, 一个索引项占 6B, 则可支持的单个文件的最大长度是多少字节? 为使单个文件的长度达到最大, 请指出起始块号和块数分别所占字节数的合理值并说明理由。
14. 某个文件系统中, 外存为硬盘。物理块大小为 512B, 有文件 A 包含 598 条记录, 每条记录占 255B, 每个物理块放 2 条记录。文件 A 所在的目录如下图所示。文件目录采用多级树形目录结构, 由根目录结点、作为目录文件的中间结点和作为信息文件的树叶组成, 每个目录项占 127B, 每个物理块放 4 个目录项, 根目录的第一块常驻内存。试问:



- 1) 若文件的物理结构采用链式存储方式, 链指针地址占 2B, 则要将文件 A 读入内存, 至少需要存取几次硬盘?
  - 2) 若文件为连续文件, 则要读文件 A 的第 487 条记录至少要存取几次硬盘?
  - 3) 一般为减少读盘次数, 可采取什么措施, 此时可减少几次存取操作?
15. 【2014 统考真题】文件 F 由 200 条记录组成, 记录从 1 开始编号。用户打开文件后, 欲将内存中的一条记录插入文件 F 中, 作为其第 30 条记录。请回答下列问题, 并说明理由。
- 1) 若文件系统采用连续分配方式, 每个磁盘块存放一条记录, 文件 F 存储区域前后均有足够的空闲磁盘空间, 则完成上述插入操作最少需要访问多少次磁盘块? F 的文件控制块内容会发生哪些改变?
  - 2) 若文件系统采用链接分配方式, 每个磁盘块存放一条记录和一个链接指针, 则完成上述插入操作需要访问多少次磁盘块? 若每个存储块大小为 1KB, 其中 4B 存放链接指针, 则该文件系统支持的文件最大长度是多少?
16. 【2018 统考真题】某文件系统采用索引结点存放文件的属性和地址信息, 簇大小为 4KB。每个文件索引结点占 64B, 有 11 个地址项, 其中直接地址项 8 个, 一级、二级和三级间接地址项各 1 个, 每个地址项长度为 4B。请回答下列问题:
- 1) 该文件系统能支持的最大文件长度是多少? (给出计算表达式即可)

- 2) 文件系统用  $1M$  ( $1M = 2^{20}$ ) 个簇存放文件索引结点，用  $512M$  个簇存放文件数据。  
若一个图像文件的大小为  $5600B$ ，则该文件系统最多能存放多少个这样的图像文件？  
3) 若文件 F1 的大小为  $6KB$ ，文件 F2 的大小为  $40KB$ ，则该文件系统获取 F1 和 F2 最后一个簇的簇号需要的时间是否相同？为什么？

### 4.2.7 答案与解析

#### 一、单项选择题

1. B

文件的物理结构包括连续、链式、索引三种，其中链式结构不能实现随机访问，连续结构的文件不易于扩展。因此随机访问且易于扩展是索引结构的特性。

2. C

每个磁盘索引块和磁盘数据块大小均为  $256B$ ，每个磁盘索引块有  $256/4 = 64$  个地址项。因此，4 个直接地址索引指向的数据块大小为  $4 \times 256B$ ；2 个一级间接索引包含的直接地址索引数为  $2 \times (256/4)$ ，即其指向的数据块大小为  $2 \times (256/4) \times 256B$ 。1 个二级间接索引所包含的直接地址索引数为  $(256/4) \times (256/4)$ ，即其所指向的数据块大小为  $(256/4) \times (256/4) \times 256B$ 。因此，7 个地址项所指向的数据块总大小为  $4 \times 256 + 2 \times (256/4) \times 256 + (256/4) \times (256/4) \times 256 = 1082368B = 1057KB$ 。

3. B

直接存取即随机存取，采用连续分配和索引分配的文件都适合于直接存取方式，只有采用链接分配的文件不具有随机存取特性。

4. A

要求有连续的存储空间，所以必须事先知道文件的大小，然后根据其大小在存储空间中找出一块大小足够的存储区。如果文件动态地增长，那么会使文件所占的空间越来越大，即使事先知道文件的最终大小，在采用预分配的存储空间的方法时，也是很无效的，它会使大量的存储空间长期闲置。

5. A

为了实现快速随机播放，要保证最短的查询时间，即不能选取链表和索引结构，因此连续结构最优。

6. B

在连续分配方式中，为了使系统能找到文件存放的地址，应在目录项的“文件物理地址”字段中，记录该文件第一条记录所在的盘块号和文件长度（以盘块数进行计量）。

7. B

顺序文件占用连续的磁盘空间，容易导致存储碎片（外部碎片）的发生。

8. C

将文件描述信息从目录项中分离，即应用了索引结点的方法，磁盘的盘块中可以存放更多的目录项，查找文件时可以大大减少其 I/O 信息量。

9. B

位示图方法是空闲块管理方法，用于管理磁盘空间。

10. A

每个磁盘块中最多有  $1KB/4B = 256$  个索引项，则两级索引分配方式下单一文件的最大长度为  $256 \times 256 \times 1KB = 64MB$ 。

11. A

四个选项中，只有 A 选项是与单个文件长度无关的。

12. C

3200 个目录项占用的盘块数为  $3200 \times 64B / 1KB = 200$  个。因为一级目录平均访盘次数为 1/2 盘块数（顺序查找目录表中的所有目录项，每个目录项为一个 FCB），所以平均的访问磁盘次数为  $200/2 = 100$  次。

13. C

选项 A 中的方法不利于对文件顺序检索，也不利于文件枚举，一般采用线性检索法；选项 B 中，为了加快文件查找速度，可以设立当前目录，于是文件路径可从当前目录进行查找；选项 D 中，在顺序检索法查找完成后，得到的是文件的逻辑地址。

14. C

文件存储空间管理即文件空闲空间管理。文件管理要解决的重要问题是，如何为创建文件分配存储空间，即如何找到空闲盘块，并对其进行管理。

15. B

块号为 100 的内存块回收时，其对应的位示图行号 row 和列号 col 分别为

$$\text{row} = (100 - 1) \text{ DIV } 32 + 1 = 4$$

$$\text{col} = (100 - 1) \text{ MOD } 32 + 1 = 4$$

即字号为 4，位号也为 4。

16. D

第 22 个逻辑记录对应 4 ( $22 \times 100 / 512 = 4$ ，余 152) 个物理块，即读入第 5 个物理块，由于文件采用的物理结构是链接文件，因此需要从目录项所指的第一个物理块开始读取，依次读到第 4 块才得到第 5 块的物理地址，共启动磁盘 5 次。修改还需要写回操作，由于写回时已获得该块的物理地址，只需 1 次访问磁盘，因此共需要启动磁盘 6 次。

17. D

通常用户可以根据需要来确定文件的逻辑结构，而文件的物理结构是由操作系统的设计师根据文件存储器的特性来确定的，一旦确定，就由操作系统管理。

18. C

打开文件表仅存放已打开文件信息的表，将指名文件的属性从外存复制到内存，再使用该文件时直接返回索引，A 错误。位图和空闲盘块链表是磁盘管理方法，B、D 错误。只有索引表中记录每个文件所存放的盘块地址，C 正确。

19. B

索引表的表项中存放有该记录的逻辑地址；三级索引需要访问 4 次磁盘；随机存取时索引文件速度快，顺序存取时顺序存取文件速度快。

20. B

10 个直接索引指针指向的数据块大小为  $10 \times 1KB = 10KB$ 。每个索引指针占 4B，则每个磁盘块可存放  $1KB / 4B = 256$  个索引指针，一级索引指针指向的数据块大小为  $256 \times 1KB = 256KB$ ，二级索引指针指向的数据块大小为  $256 \times 256 \times 1KB = 2^{16}KB = 64MB$ 。

按字节编址，偏移量为 1234 时，因  $1234B < 10KB$ ，由直接索引指针可得到其所在的磁盘块地址。文件的索引结点已在内存中，因此地址可直接得到，因此仅需 1 次访盘即可。

偏移量为 307400 时，因  $10KB + 256KB < 307400B < 64MB$ ，可知该偏移量的内容在二级索引指针所指向的某个磁盘块中，索引结点已在内存中，因此先访盘 2 次得到文件所在的磁盘块地址，再访盘 1 次即可读出内容，共需 3 次访盘。

21. C

盘块号 = 起始块号 +  $\lfloor \text{盘块号}/(1024 \times 8) \rfloor = 32 + \lfloor 409612/(1024 \times 8) \rfloor = 32 + 50 = 82$ ，这里问的是块内字节号而不是位号，因此还需除以 8 (1B = 8 位)，块内字节号 =  $\lfloor (\text{盘块号} \% (1024 \times 8))/8 \rfloor = 1$ 。

22. B

传统文件系统管理空闲磁盘的方法包括空闲表法、空闲链表法、位示图法和成组链接法，I、III 正确。文件分配表 (FAT) 的表项与物理磁盘块一一对应，并且可以用一个特殊的数字 -1 表示文件的最后一块，用 -2 表示这个磁盘块是空闲的（当然，规定用 -3, -4 来表示也是可行的）。因此文件分配表 (FAT) 不仅记录了文件中各个块的先后链接关系，同时还标记了空闲的磁盘块，操作系统可以通过 FAT 对文件存储空间进行管理，IV 正确。索引结点是操作系统为了实现文件名与文件信息分开而设计的数据结构，存储了文件描述信息，索引结点属于文件目录管理部分的内容，II 错误。

23. A

索引分配支持变长的文件，同时可以随机访问文件的指定数据块，A 正确。链接分配不支持随机访问，需要依靠指针依次访问，B 错误。连续分配的文件长度固定，不支持可变文件长度（连续分配的文件长度虽然也可变，但是需要大量移动数据，代价较大，相比之下不太合适），C 错误。动态分区分配是内存管理方式，不是磁盘空间的管理方式，D 错误。

24. B

在总长为 64 字节的目录项中，索引结点占 4 字节，即 32 位。不同目录下的文件的文件名可以相同，所以在考虑系统创建最多文件数量时，只需考虑索引结点的个数，即创建文件数量上限 = 索引结点数量上限。整个系统中最多存储  $2^{32}$  个索引结点，因此整个系统最多可以表示  $2^{32}$  个文件，B 正确。

## 二、综合应用题

1. 解答：

连续分配方式的优点是可以随机访问（磁盘），访问速度快；缺点是要求有连续的存储空间，容易产生碎片，降低磁盘空间利用率，并且不利于文件的增长扩充。

链接分配方式的优点是不要求连续的存储空间，能更有效地利用磁盘空间，并且有利于扩充文件；缺点是只适合顺序访问，不适合随机访问；另外，链接指针占用一定的空间，降低了存储效率，可靠性也差。

索引分配方式的优点是既支持顺序访问又支持随机访问，查找效率高，便于文件删除；缺点是索引表会占用一定的存储空间。

2. 解答：

“FCB 分解法”加快目录检索速度的原理是：目录是存在磁盘上的，所以检索目录时需要访问磁盘，速度很慢；而 FCB 分解法将 FCB 的一部分数据分解出去，存放在另一个数据结构中，在目录中仅留下文件的基本信息和指向该数据结构的指针，这样一来就有效地缩减了目录的体积，减少了目录所占磁盘的块数，检索目录时读取磁盘的次数也减少，于是就加快了检索目录的次数。

因为原本整个 FCB 都是在目录中的，而 FCB 分解法将 FCB 的部分内容放在了目录外，所以检索完目录后还需要读取一次磁盘，以找齐 FCB 的所有内容。

1) 分解法前，目录的磁盘块数为  $64 \times 254 / 512 = 31.75$ ，即 32 块。

所找目录项在第 1, 2, 3, …, 32 块所需的磁盘访问次数分别为 1, 2, 3, …, 32 次。所以查找该目录文件的某个 FCB 的平均访问磁盘次数 =  $(1 + 2 + 3 + \dots + 32) / 32 = 16.5$  次。

分解法后，目录的磁盘块数为  $10 \times 254 / 512 = 4.96$ ，即 5 块。

所找目录项在第 1, 2, 3, 4, 5 块所需的磁盘访问次数分别为 2, 3, 4, 5, 6 次。所以查找该目录文件的某个 FCB 的平均访问磁盘次数  $= (2 + 3 + 4 + 5 + 6) / 5 = 4$  次。

2) 分解法前平均访问磁盘次数  $= (1 + 2 + 3 + \dots + n) / n = n \times (n + 1) / 2n = (n + 1) / 2$  次。

分解法后平均访问磁盘次数  $= (2 + 3 + 4 + \dots + (m + 1)) / m = m \times (m + 3) / 2m = (m + 3) / 2$  次。

为了使访问磁盘次数减少，显然需要  $(m + 3) / 2 < (n + 1) / 2$ ，即  $m < n - 2$ 。

3. 解答：

因为  $1569 = 512 \times 3 + 33$ ，所以要访问字节的逻辑记录号为 3，对应的物理磁盘块号为 80，因此应访问第 80 号磁盘块。

4. 解答：

1) 在磁盘中连续存放（采取连续结构），磁盘寻道时间更短，文件随机访问效率更高；在 FCB 中加入的字段为<起始块号，块数>或<起始块号，结束块号>。

2) 将所有的 FCB 集中存放，文件数据集中存放。这样在随机查找文件名时，只需访问 FCB 对应的块，可减少磁头移动和磁盘 I/O 访问次数。

5. 解答：

对于 540MB 的硬盘，硬盘总块数为  $540MB / 1KB = 540K$  个。

因为  $540K$  刚好小于  $2^{20}$ ，所以文件分配表的每个表目可用 20 位，即  $20 / 8 = 2.5B$ ，这样 FAT 占用的存储空间大小为  $2.5B \times 540K = 1350KB$ 。

6. 解答：

1) 两个目录文件 dir 和 dir1 的内容如下表所示。

dir 目录文件		dir1 目录文件	
文件名	簇号	文件名	簇号
dir1	48	file1	100
		file2	200

2) FAT 的最大长度为  $2^{16} \times 2B = 128KB$ ，文件的最大长度是  $2^{16} \times 4KB = 256MB$ 。

3) file1 的簇号 106 存放在 FAT 的 100 号表项中，簇号 108 存放在 FAT 的 106 号表项中。

4) 需要访问目录文件 dir1 所在的 48 号簇及文件 file1 的 106 号簇。

7. 解答：

1) 因为磁盘块大小为 512B，所以索引块大小也为 512B，每个磁盘地址大小为 2B。因此，一个一级索引表可容纳 256 个磁盘地址。同样，一个二级索引表可容纳 256 个一级索引表地址，一个三级索引表可容纳 256 个二级索引表地址。这样，一个普通文件最多可有的文件页数为  $10 + 256 + 256 \times 256 + 256 \times 256 \times 256 = 16843018$ 。

2) 由图可知，目录文件 A 和 D 中的目录项都只有两个，因此这两个目录文件都只占用一个物理块。要读文件 J 中的某一页，先从内存的根目录中找到目录文件 A 的磁盘地址，将其读入内存（已访问磁盘 1 次）。然后从目录 A 中找出目录文件 D 的磁盘地址读入内存（已访问磁盘 2 次）。再从目录 D 中找出文件 J 的 FCB 地址读入内存（已访问磁盘 3 次）。在最坏情况下，该访问页存放在三级索引下，这时候需要一级级地读三级索引块才能得到文件 J 的地址（已访问磁盘 6 次）。最后读入文件 J 中的相应页（共访问磁盘 7 次）。所以，若要读文件 J 中的某一页，最多启动磁盘 7 次。

3) 由图可知，目录文件 C 和 U 的目录项较多，可能存放在多个链接在一起的磁盘块中。在

最好情况下，所需的目录项都在目录文件的第一个磁盘块中。先从内存的根目录中找到目录文件 C 的磁盘地址并读入内存（已访问磁盘 1 次）。在 C 中找出目录文件 I 的磁盘地址并读入内存（已访问磁盘 2 次）。在 I 中找出目录文件 P 的磁盘地址并读入内存（已访问磁盘 3 次）。从 P 中找到目录文件 U 的磁盘地址并读入内存（已访问磁盘 4 次）。从 U 的第一个磁盘块中找出文件 W 的 FCB 地址并读入内存（已访问磁盘 5 次）。在最好情况下，要访问的页在 FCB 的前 10 个直接块中，按照直接块指示的地址读文件 W 的相应页（已访问磁盘 6 次）。所以，若要读文件 W 中的某页，最少启动磁盘 6 次。

- 4) 为了减少启动磁盘的次数，可以将需要访问的 W 文件挂在根目录的最前面的目录项中。此时，只需读内存中的根目录就可找到 W 的 FCB，将 FCB 读入内存（已访问磁盘 1 次），最差情况下，需要的 W 文件的那个页挂在 FCB 的三级索引下，因此读 3 个索引块需要访问磁盘 3 次（已访问磁盘 4 次）得到该页的物理地址，再去读这个页即可（已访问磁盘 5 次）。此时，磁盘最多启动 5 次。

#### 8. 解答：

- 1) 要想只用到索引结点的直接块，这个文件应能全部在 10 个直接块指向的数据块中放下，而数据块的大小为 4KB，所以该文件的大小应小于等于  $4KB \times 10 = 40KB$ ，即文件的大小不超过 40KB 时可以只用到索引结点的直接块。
- 2) 只需要算出索引结点指向的所有数据块的块数，再乘以数据块的大小即可。直接块指向的数据块数 = 10 块。

一级间接块指向的索引块里的指针数为  $4KB/4B = 1024$ ，所以一级间接块指向的数据块数为 1024 块。

二级间接块指向的索引块里的指针数为  $4KB/4B = 1024$ ，指向的索引块里再拥有  $4KB/4B = 1024$  个指针数。所以二级间接块指向的数据块数为  $(4KB/4B)^2 = 1024^2$ 。

三级间接块指向的数据块数为  $(4KB/4B)^3 = 1024^3$ 。所以，该索引结点能访问到的地址空间大小为

$$\left[ 10 + 1 \times \frac{4KB}{4B} + 1 \times \left( \frac{4KB}{4B} \right)^2 + 1 \times \left( \frac{4KB}{4B} \right)^3 \right] \times 4KB \approx 4100.00GB = 4.00TB$$

- 3) 因为  $10000B/4KB = 2.44$ ，所以第 10000B 的内容存放在第 3 个直接块中，若要读取一个文件的第 10000B 的内容，需要访问磁盘 1 次。
- 4) 因为 10MB 的内容需要数据块数为  $10MB/4KB = 2.5 \times 1024$ ，直接块和一级间接块指向的数据块数 =  $10 + (4KB/4B) = 1034 < 2.5 \times 1024$ ，直接块和一级间接块及二级间接块的数据块数 =  $10 + (4KB/4B) + (4KB/4B)^2 > 1 \times 1024^2 > 2.5 \times 1024$ ，所以第 10MB 数据应该在二级间接块下属的某个数据块中，若要读取一个文件的第 10MB 的内容，需要访问磁盘 3 次。

#### 9. 解答：

第一问要计算混合索引结构的寻址空间大小；第二问只要计算出存储该文件索引块的大小，然后加上该文件本身的大小即可。

- 1) 物理块大小为 4KB，数据大小为 4B，则每个物理块可存储的地址数为  $4KB/4B = 1024$ 。最大文件的物理块数可达  $10 + 1024 + 1024^2 + 1024^3$ ，每个物理块大小为 4KB，因此总长度为

$$(10 + 1024 + 1024^2 + 1024^3) \times 4KB = 40KB + 4MB + 4GB + 4TB$$

这个文件系统允许的最大文件长度是 4TB + 4GB + 4MB + 40KB，约为 4TB。

- 2) 占用空间分为文件实际大小和索引项大小，文件大小为 2GB，从 1) 中的计算知，需要

使用到二次间接索引项。该文件占用  $2\text{GB}/4\text{KB} = 512 \times 1024$  个数据块。

一次间接索引项使用 1 个间接索引块，二次间接索引项使用  $1 + \lceil(512 \times 1024 - 10 - 1024)/1024\rceil \approx 512$  个间接索引块（最左的 1 表示二次间址块），所以间接索引块所占空间大小为

$$(1 + 512) \times 4\text{KB} = 2\text{MB} + 4\text{KB}$$

另外每个文件使用的 *i\_node* 数据结构占  $13 \times 4\text{B} = 52\text{B}$ ，因此该文件实际占用磁盘空间大小为  $2\text{GB} + 2\text{MB} + 4\text{KB} + 52\text{B}$ 。

#### 10. 解答：

- 1) 根据位示图的位置  $(i, j)$ ，得出盘块的序号  $b = i \times 16 + j$ ；用  $C$  表示柱面号， $H$  表示磁头号， $S$  表示扇区号，则有

$$C = b/(20 \times 8), \quad H = (b \% (20 \times 8))/8, \quad S = b \% 8$$

- 2) 分配：顺序扫描位示图，找出 1 个其值为“0”的二进制位（“0”表示空闲），利用上述公式将其转换成相应的序号  $b$ ，并修改位示图，置  $(i, j) = 1$ 。

回收：将回收盘块的盘块号换算成位示图中的  $i$  和  $j$ ，转换公式为

$$b = C \times 20 \times 8 + H \times 8 + S, \quad i = b/16, \quad j = b \% 16$$

最后将计算出的  $(i, j)$  在位示图中置“0”。

#### 11. 解答：

- 1) 整个磁盘空间的存储块数目为  $4 \times 16 \times 100 = 6400$  个。  
2) 位示图应为 6400 个位，如果用字长为 32 位（即  $n=32$ ）的单元来构造位示图，那么需要  $6400/32 = 200$  个字。

- 3) 位示图中第 18 个字的第 16 位（即  $i=18, j=16$ ）对应的块号为  $32 \times (18-1) + 16 = 560$ 。

#### 12. 解答：

由于块长为 512B，每个块号长 2B，因此一个一级索引表可容纳 256 个磁盘块地址。同样，一个二级索引表可容纳 256 个磁盘块地址，一个三级索引表也可容纳 256 个磁盘块地址。

所以采用二级索引时，可寻址的文件最大长度是

$$256 \times 256 \times 512 = 32\text{MB}$$

采用三级索引时，可寻址的文件最大长度是

$$256 \times 256 \times 256 \times 512 = 8\text{GB}$$

#### 13. 解答：

- 1) 文件系统中所能容纳的磁盘块总数为  $4\text{TB}/1\text{KB} = 2^{32}$ 。要完全表示所有磁盘块，索引项中的块号最少要占  $32/8 = 4\text{B}$ 。而索引表区仅采用直接索引结构，因此 512B 的索引表区能容纳  $512\text{B}/4\text{B} = 128$  个索引项。每个索引项对应一个磁盘块，所以该系统可支持的单个文件最大长度是  $128 \times 1\text{KB} = 128\text{KB}$ 。

- 2) 这里考查的分配方式不同于我们熟悉的三种经典分配方式，但题目中给出了详细的解释。所求的单个文件最大长度一共包含两部分：预分配的连续空间和直接索引区。

连续区块数占 2B，共可表示  $2^{16}$  个磁盘块，即  $2^{26}\text{B}$ 。直接索引区共  $504\text{B}/6\text{B} = 84$  个索引项。所以该系统可支持的单个文件最大长度是  $2^{26}\text{B} + 84\text{KB}$ 。

为了使单个文件的长度达到最大，应使连续区的块数字段表示的空间大小尽可能接近系统最大容量 4TB。分别设起始块号和块数占 4B，这样起始块号可以寻址的范围是  $2^{32}$  个磁盘块，共 4TB，即整个系统空间。同样，块数字段可以表示最多  $2^{32}$  个磁盘块，共 4TB。

#### 14. 解答：

- 1) 由于根目录的第一块常驻内存（即 *root* 所指的 /bin, /dev, /etc, /boot 等可直接获得），根目

录找到文件 A 需要 5 次读盘。由  $255 \times 2 + 2 = 512$  可知，一个物理块在链式存储结构下可放 2 条记录及下一个物理块地址，而文件 A 共有 598 条记录，因此读取 A 的所有记录所需的读盘次数为  $598/2 = 299$ ，所以将文件 A 读到内存至少需读盘  $299 + 5 = 304$  次。

- 2) 当文件为连续文件时，找到文件 A 同样需要 5 次读盘，且知道文件 A 的地址后通过计算只需一次读盘即可读出第 487 条记录，所以至少需要  $5 + 1 = 6$  次读盘。
- 3) 为减少因查找目录而读盘的次数，可采用索引结点方法。若一个目录项占 16B，则一个盘块可存放  $512/16 = 32$  个目录项，与本题一个盘块仅能存放 4 个目录相比，可使因访问目录而读盘的次数减少  $1/8$ 。对查找文件的记录而言，可用一个或多个盘块来存放该文件的所有盘块号，即用链接索引方法；一个盘块可存放  $512/2 - 1 = 255$  个盘块号，留下一个地址用来指向下一个存储盘块号（索引块）的磁盘块号。这样，就本题来说，查找目录时需启动 5 次磁盘。文件 A 共有 299 个盘块，则查找文件 A 的某一记录时需两次取得所有盘块号，再需最多启动一次磁盘即可把 A 中的任意一条记录读入内存。所以，查找一条记录最多需要 8 次访盘，而原来的链接方法查找一条记录时，读盘次数为 6~304。

#### 15. 解答：

- 1) 系统采用顺序分配方式时，插入记录需要移动其他的记录块，整个文件共有 200 条记录，要插入新记录作为第 30 条，而存储区前后均有足够的磁盘空间，且要求最少的访问存储块数，则要把文件前 29 条记录前移，若算访盘次数，移动一条记录读出和存回磁盘各是一次访盘，29 条记录共访盘 58 次，存回第 30 条记录访盘 1 次，共访盘 59 次。  
F 的文件控制区的起始块号和文件长度的内容会因此改变。
- 2) 文件系统采用链接分配方式时，插入记录并不用移动其他记录，只需找到相应的记录，修改指针即可。插入的记录为其第 30 条记录，因此需要找到文件系统的第 29 块，一共需要访盘 29 次，然后把第 29 块的下块地址部分赋给新块，把新块存回磁盘会访盘 1 次，然后修改内存中第 29 块的下块地址字段，再存回磁盘，一共访盘 31 次。  
4B 共 32 位，可以寻址  $2^{32} = 4G$  块存储块，每块的大小为 1KB，即 1024B，其中下块地址部分占 4B，数据部分占 1020B，因此该系统的文件最大长度是  $4G \times 1020B = 4080GB$ 。

#### 16. 解答：

- 1) 簇大小为 4KB，每个地址项长度为 4B，因此每簇有  $4KB/4B = 1024$  个地址项。最大文件的物理块数可达  $8 + 1 \times 1024 + 1 \times 1024^2 + 1 \times 1024^3$ ，每个物理块（簇）大小为 4KB，因此最大文件长度为  $(8 + 1 \times 1024 + 1 \times 1024^2 + 1 \times 1024^3) \times 4KB = 32KB + 4MB + 4GB + 4TB$ 。
- 2) 文件索引结点总个数为  $1M \times 4KB/64B = 64M$ ，5600B 的文件占 2 个簇，512M 个簇可存放的文件总个数为  $512M/2 = 256M$ 。可表示的文件总个数受限于文件索引结点总个数，因此能存储 64M 个大小为 5600B 的图像文件
- 3) 文件 F1 的大小为  $6KB < 4KB \times 8 = 32KB$ ，因此获取文件 F1 的最后一个簇的簇号只需要访问索引结点的直接地址项。文件 F2 大小为 40KB， $4KB \times 8 < 40KB < 4KB \times 8 + 4KB \times 1024$ ，因此获取 F2 的最后一个簇的簇号还需要读一级索引表。综上，需要的时间不相同。

## 4.3 磁盘组织与管理

在学习本节时，请读者思考以下问题：

- 1) 在磁盘上进行一次读写操作需要哪几部分时间？其中哪部分时间最长？

2) 存储一个文件时, 当一个磁道存储不下时, 剩下部分是存在同一个盘面的不同磁道好, 还是存在同一个柱面上的不同盘面好?

本节主要介绍文件系统管理磁盘的方式, 由于内容较少且属于实现部分, 不需要问题来引导学习, 因此本节不给出问题。学习本节时, 要重点掌握计算一次磁盘操作的时间, 以及对于给定访盘的磁道序列, 按照特定算法求出磁头通过的总磁道数及平均寻道数。

### 4.3.1 磁盘的结构

磁盘 (Disk) 是由表面涂有磁性物质的金属或塑料构成的圆形盘片, 通过一个称为磁头的导体线圈从磁盘存取数据。在读/写操作期间, 磁头固定, 磁盘在下面高速旋转。如图 4.17 所示, 磁盘盘面上的数据存储在一组同心圆中, 称为磁道。每个磁道与磁头一样宽, 一个盘面有上千个磁道。磁道又划分为几百个扇区, 每个扇区固定存储大小 (通常为 512B), 一个扇区称为一个盘块。相邻磁道及相邻扇区间通过一定的间隙分隔开, 以避免精度错误。注意, 由于扇区按固定圆心角度划分, 所以密度从最外道向里道增加, 磁盘的存储能力受限于最内道的最大记录密度。

磁盘安装在一个磁盘驱动器中, 它由磁头臂、用于旋转磁盘的主轴和用于数据输入/输出的电子设备组成。如图 4.18 所示, 多个盘片垂直堆叠, 组成磁盘组, 每个盘面对应一个磁头, 所有磁头固定在一起, 与磁盘中心的距离相同且一起移动。所有盘片上相对位置相同的磁道组成柱面。按照这种物理结构组织, 扇区就是磁盘可寻址的最小存储单位, 磁盘地址用“柱面号·盘面号·扇区号 (或块号)”表示。

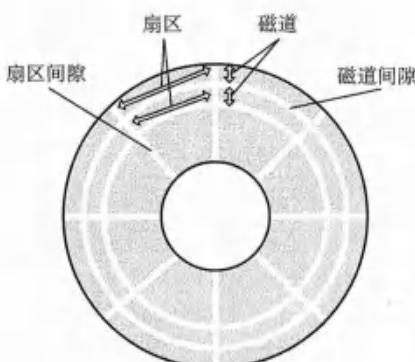


图 4.17 磁盘盘片

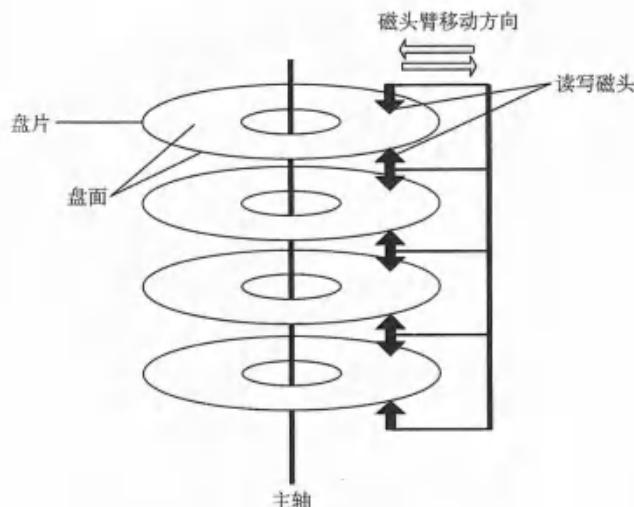


图 4.18 磁盘驱动器

磁盘按不同的方式可分为若干类型: 磁头相对于盘片的径向方向固定的, 称为固定头磁盘, 每个磁道一个磁头; 磁头可移动的, 称为活动头磁盘, 磁头臂可来回伸缩定位磁道; 磁盘永久固定在磁盘驱动器内的, 称为固定盘磁盘; 可移动和替换的, 称为可换盘磁盘。

前面说过, 操作系统中几乎每介绍一类资源及对这类资源的管理时, 都要涉及一类调度算法。用户访问文件, 需要操作系统的服务, 文件实际上存储在磁盘中, 操作系统接收用户的命令后, 经过一系列的检验访问权限和寻址过程后, 最终都会到达磁盘, 控制磁盘把相应的数据信息读出或修改。当有多个请求同时到达时, 操作系统就要决定先为哪个请求服务, 这就是磁盘调度算法要解决的问题。

### 4.3.2 磁盘调度算法

一次磁盘读写操作的时间由寻找(寻道)时间、旋转延迟时间和传输时间决定。

- 1) 寻找时间  $T_s$ 。活动头磁盘在读写信息前, 将磁头移动到指定磁道所需要的时间。这个时间除跨越  $n$  条磁道的时间外, 还包括启动磁臂的时间  $s$ , 即

$$T_s = m \times n + s$$

式中,  $m$  是与磁盘驱动器速度有关的常数, 约为  $0.2\text{ms}$ , 磁臂的启动时间为  $2\text{ms}$ 。

- 2) 旋转延迟时间  $T_r$ 。磁头定位到某一磁道的扇区所需要的时间, 设磁盘的旋转速度为  $r$ , 则

$$T_r = \frac{1}{2r}$$

对于硬盘, 典型的旋转速度为 5400 转/分, 相当于一周  $11.1\text{ms}$ , 则  $T_r$  为  $5.55\text{ms}$ ; 对于软盘, 其旋转速度为 300~600 转/分, 则  $T_r$  为  $50\sim100\text{ms}$ 。

- 3) 传输时间  $T_t$ 。从磁盘读出或向磁盘写入数据所经历的时间, 这个时间取决于每次所读/写的字节数  $b$  和磁盘的旋转速度:

$$T_t = \frac{b}{rN}$$

式中,  $r$  为磁盘每秒的转数,  $N$  为一个磁道上的字节数。

在磁盘存取时间的计算中, 寻道时间与磁盘调度算法相关, 下面将介绍分析几种算法; 而延迟时间和传输时间都与磁盘旋转速度相关, 且为线性相关, 所以在硬件上, 转速是磁盘性能的一个非常重要的参数。

总平均存取时间  $T_a$  可以表示为

$$T_a = T_s + \frac{1}{2r} + \frac{b}{rN}$$

虽然这里给出了总平均存取时间的公式, 但是这个平均值是没有太大实际意义的, 因为在实际的磁盘 I/O 操作中, 存取时间与磁盘调度算法密切相关。调度算法直接决定寻找时间, 从而决定总的存取时间。

目前常用的磁盘调度算法有以下几种。

#### (1) 先来先服务 (First Come First Served, FCFS) 算法

FCFS 算法根据进程请求访问磁盘的先后顺序进行调度, 这是一种最简单的调度算法, 如图 4.19 所示。该算法的优点是具有公平性。若只有少量进程需要访问, 且大部分请求都是访问簇聚的文件扇区, 则有望达到较好的性能; 若有大量进程竞争使用磁盘, 则这种算法在性能上往往接近于随机调度。所以, 实际磁盘调度中会考虑一些更为复杂的调度算法。

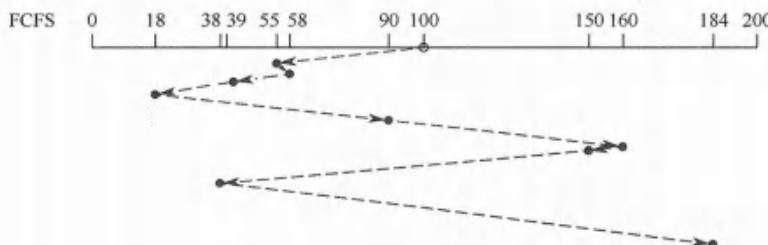


图 4.19 FCFS 磁盘调度算法

例如, 磁盘请求队列中的请求序列为  $55, 58, 39, 18, 90, 160, 150, 38, 184$ , 磁头的初始位

置是磁道 100，采用 FCFS 算法时磁头的运动过程如图 4.19 所示。磁头共移动了  $(45 + 3 + 19 + 21 + 72 + 70 + 10 + 112 + 146) = 498$  个磁道，平均寻找长度  $= 498/9 = 55.3$ 。

#### (2) 最短寻找时间优先 (Shortest Seek Time First, SSTF) 算法

SSTF 算法选择调度处理的磁道是与当前磁头所在磁道距离最近的磁道，以便使每次的寻找时间最短。当然，总是选择最小寻找时间并不能保证平均寻找时间最小，但能提供比 FCFS 算法更好的性能。这种算法会产生“饥饿”现象。如图 4.20 所示，若某时刻磁头正在 18 号磁道，而在 18 号磁道附近频繁地增加新的请求，则 SSTF 算法使得磁头长时间在 18 号磁道附近工作，将使 184 号磁道的访问被无限期地延迟，即被“饿死”。

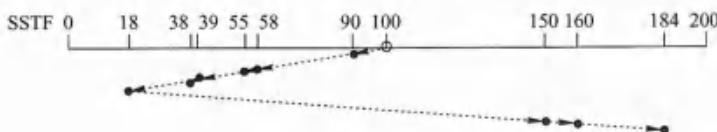


图 4.20 SSTF 磁盘调度算法

例如，磁盘请求队列中的请求顺序分别为 55, 58, 39, 18, 90, 160, 150, 38, 184，磁头初始位置是磁道 100，采用 SSTF 算法时磁头的运动过程如图 4.20 所示。磁头共移动了  $10 + 32 + 3 + 16 + 1 + 20 + 132 + 10 + 24 = 248$  个磁道，平均寻找长度  $= 248/9 = 27.5$ 。

#### (3) 扫描 (SCAN) 算法 (又称电梯调度算法)

SCAN 算法在磁头当前移动方向上选择与当前磁头所在磁道距离最近的请求作为下一次服务的对象，实际上就是在最短寻找时间优先算法的基础上规定了磁头运动的方向，如图 4.21 所示。由于磁头移动规律与电梯运行相似，因此又称电梯调度算法。SCAN 算法对最近扫描过的区域不公平，因此它在访问局部性方面不如 FCFS 算法和 SSTF 算法好。

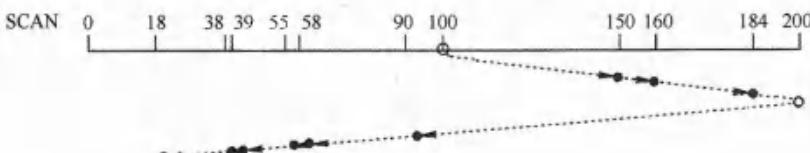


图 4.21 SCAN 磁盘调度算法

例如，磁盘请求队列中的请求顺序分别为 55, 58, 39, 18, 90, 160, 150, 38, 184，磁头初始位置是磁道 100。采用 SCAN 算法时，不但要知道磁头的当前位置，而且要知道磁头的移动方向，假设磁头沿磁道号增大的顺序移动，则磁头的运动过程如图 4.21 所示。移动磁道的顺序为 100, 150, 160, 184, 200, 90, 58, 55, 39, 38, 18。磁头共移动了  $(50 + 10 + 24 + 16 + 110 + 32 + 3 + 16 + 1 + 20) = 282$  个磁道，平均寻道长度  $= 282/9 = 31.33$ 。

#### (4) 循环扫描 (Circular SCAN, C-SCAN) 算法

在扫描算法的基础上规定磁头单向移动来提供服务，回返时直接快速移动至起始端而不服务任何请求。由于 SCAN 算法偏向于处理那些接近最里或最外的磁道的访问请求，所以使用改进型的 C-SCAN 算法来避免这个问题，如图 4.22 所示。

采用 SCAN 算法和 C-SCAN 算法时，磁头总是严格地遵循从盘面的一端到另一端，显然，在实际使用时还可以改进，即磁头移动只需要到达最远端的一个请求即可返回，不需要到达磁盘端点。这种形式的 SCAN 算法和 C-SCAN 算法称为 LOOK 调度 (见图 4.23-1) 和 C-LOOK (见图 4.23-2) 调度，因为它们在朝一个给定方向移动前会查看是否有请求。

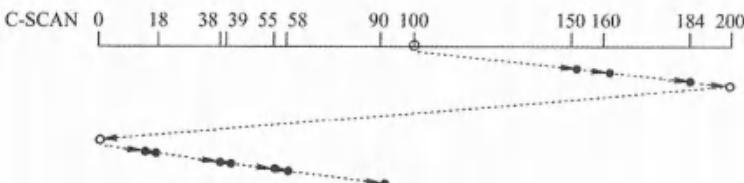


图 4.22 C-SCAN 磁盘调度算法

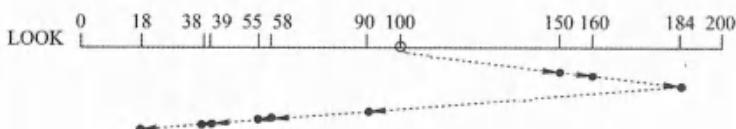


图 4.23-1 LOOK 磁盘调度算法

注意，若无特别说明，也可以默认 SCAN 算法和 C-SCAN 算法为 LOOK 和 C-LOOK 调度（请读者认真领悟，并通过结合后面的习题进一步加深对以上相关算法的理解）。

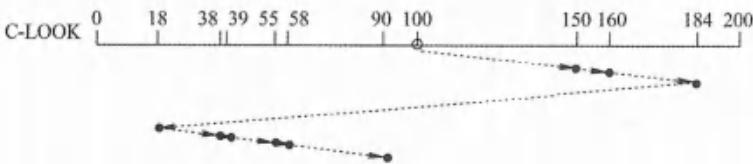


图 4.23-2 C-LOOK 磁盘调度算法

例如，磁盘请求队列中的请求顺序为 55, 58, 39, 18, 90, 160, 150, 38, 184，磁头初始位置是磁道 100。采用 C-SCAN 算法时，假设磁头沿磁道号增大的顺序移动，则磁头的运动过程如图 4.23 所示。移动磁道的顺序为 100, 150, 160, 184, 200, 0, 18, 38, 39, 55, 58, 90。磁头共移动  $50 + 10 + 24 + 16 + 200 + 18 + 20 + 1 + 16 + 3 + 32 = 390$  个磁道，平均寻道长度  $= 390/9 = 43.33$ 。

不太熟悉操作系统整体框架的读者经常混淆磁盘调度算法中的循环扫描算法和页面调度算法中的 CLOCK 算法，请读者注意区分。

对比以上几种磁盘调度算法，FCFS 算法太过简单，性能较差，仅在请求队列长度接近于 1 时才较为理想；SSTF 算法较为通用和自然；SCAN 算法和 C-SCAN 算法在磁盘负载较大时比较占优势。它们之间的比较见表 4.4。

表 4.4 磁盘调度算法比较

	优 点	缺 点
FCFS 算法	公平、简单	平均寻道距离大，仅应用在磁盘 I/O 较少的场合
SSTF 算法	性能比“先来先服务”好	不能保证平均寻道时间最短，可能出现“饥饿”现象
SCAN 算法	寻道性能较好，可避免“饥饿”现象	不利于远离磁头一端的访问请求
C-SCAN 算法	消除了对两端磁道请求的不公平	—

除减少寻找时间外，减少延迟时间也是提高磁盘传输效率的重要因素。可以对盘面扇区进行交替编号，对磁盘片组中的不同盘面错位命名。假设每个盘面有 8 个扇区，磁盘片组共 8 个盘面，则可以采用如图 4.24 所示的编号。

磁盘是连续自转设备，磁头读/写一个物理块后，需要经过短暂的处理时间才能开始读/写下一块。假设逻辑记录数据连续存放在磁盘空间中，若在盘面上按扇区交替编号连续存放，则连续

读/写多条记录时能减少磁头的延迟时间；同柱面不同盘面的扇区若能错位编号，连续读/写相邻两个盘面的逻辑记录时也能减少磁头延迟时间。

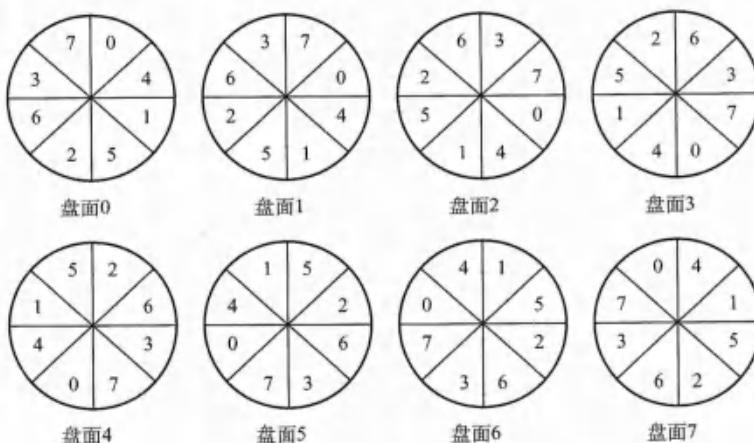


图 4.24 磁盘片组扇区编号

以图 4.24 为例，在随机扇区访问情况下，定位磁道中的一个扇区平均需要转过 4 个扇区，这时，延迟时间是传输时间的 4 倍，这是一种非常低效的存取方式。理想化的情况是不需要定位而直接连续读取扇区，没有延迟时间，这样磁盘数据存取效率可以成倍提高。但由于读取扇区的顺序是不可预测的，所以延迟时间不可避免。图 4.24 中的编号方式是读取连续编号扇区时的一种方法。

磁盘寻块时间分为三个部分，即寻道时间、延迟时间和传输时间，寻道时间和延迟时间属于“找”的时间，凡是“找”的时间都可以通过一定的方法削减，但传输时间是磁盘本身性质所决定的，不能通过一定的措施减少。

### 4.3.3 磁盘的管理

#### 1. 磁盘初始化

一个新的磁盘只是一个含有磁性记录材料的空白盘。在磁盘能存储数据之前，它必须分成扇区以便磁盘控制器能进行读和写操作，这个过程称为低级格式化（物理分区）。低级格式化为磁盘的每个扇区采用特别的数据结构。每个扇区的数据结构通常由头、数据区域（通常为 512B 大小）和尾部组成。头部和尾部包含了一些磁盘控制器所使用的信息。

为了使用磁盘存储文件，操作系统还需要将自己的数据结构记录在磁盘上：第一步将磁盘分为由一个或多个柱面组成的分区（即我们熟悉的 C 盘、D 盘等形式的分区）；第二步对物理分区进行逻辑格式化（创建文件系统），操作系统将初始的文件系统数据结构存储到磁盘上，这些数据结构包括空闲和已分配的空间及一个初始为空的目录。

#### 2. 引导块

计算机启动时需要运行一个初始化程序（自举程序），它初始化 CPU、寄存器、设备控制器和内存等，接着启动操作系统。为此，该自举程序应找到磁盘上的操作系统内核，装入内存，并转到起始地址，从而开始操作系统的运行。

自举程序通常保存在 ROM 中，为了避免改变自举代码而需要改变 ROM 硬件的问题，因此只在 ROM 中保留很小的自举装入程序，将完整功能的自举程序保存在磁盘的启动块上，启动块

位于磁盘的固定位。拥有启动分区的磁盘称为启动磁盘或系统磁盘。

### 3. 坏块

由于磁盘有移动部件且容错能力弱，因此容易导致一个或多个扇区损坏。部分磁盘甚至从出厂时就有坏扇区。根据所使用的磁盘和控制器，对这些块有多种处理方式。

对于简单磁盘，如电子集成驱动器（IDE），坏扇区可手工处理，如 MS-DOS 的 Format 命令执行逻辑格式化时便会扫描磁盘以检查坏扇区。坏扇区在 FAT 表上会标明，因此程序不会使用。

对于复杂的磁盘，如小型计算机系统接口（SCSI），其控制器维护一个磁盘坏块链表。该链表在出厂前进行低级格式化时就已初始化，并在磁盘的整个使用过程中不断更新。低级格式化将一些块保留作为备用，对操作系统透明。控制器可用备用块来逻辑地替代坏块，这种方案称为扇区备用。

对坏块的处理实质上就是用某种机制，使系统不去使用坏块。坏块属于硬件故障，操作系统是不能修复坏块的。

## 4.3.4 本节小结

本节开头提出的问题的参考答案如下。

1) 在磁盘上进行一次读写操作需要哪几部分时间？其中哪部分时间最长？

在磁盘上进行一次读写操作花费的时间由寻道时间、延迟时间和传输时间决定。其中寻道时间是将磁头移动到指定磁道所需要的时间，延迟时间是磁头定位到某一磁道的扇区（块号）所需要的时间，传输时间是从磁盘读出或向磁盘写入数据所经历的时间。一般来说，寻道时间因为要移动磁臂，所以占用时间最长。

2) 存储一个文件时，当一个磁道存储不下时，剩下部分是存在同一个盘面的不同磁道好，还是存在同一个柱面上的不同盘面好？

上一问已经说到，寻道时间对于一次磁盘访问的影响是最大的，若存在同一个盘面的不同磁道，则磁臂势必要移动，这样会大大增加文件的访问时间，而存在同一个柱面上的不同盘面就不需要移动磁道，所以一般情况下存在同一个柱面上的不同盘面更好。

## 4.3.5 本节习题精选

### 一、单项选择题

1. 磁盘是可共享设备，但在每个时刻（ ）作业启动它。
 

A. 可以由任意多个	B. 能限定多个
C. 至少能由一个	D. 至多能由一个
2. 用磁带做文件存储介质时，文件只能组织成（ ）。
 

A. 顺序文件	B. 链接文件	C. 索引文件	D. 目录文件
---------	---------	---------	---------
3. 既可以随机访问又可顺序访问的有（ ）。
 

I. 光盘	II. 磁带	III. U 盘	IV. 磁盘
A. II、III、IV	B. I、III、IV	C. III、IV	D. 仅 IV
4. 磁盘调度的目的是缩短（ ）时间。
 

A. 找道	B. 延迟	C. 传送	D. 启动
-------	-------	-------	-------
5. 磁盘上的文件以（ ）为单位读/写。
 

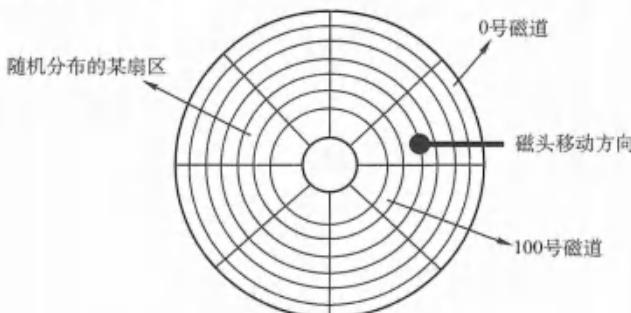
A. 块	B. 记录	C. 柱面	D. 磁道
------	-------	-------	-------

6. 在磁盘中读取数据的下列时间中，影响最大的是（ ）。
- 处理时间
  - 延迟时间
  - 传送时间
  - 寻找时间
7. 在下列有关旋转延迟的叙述中，不正确的是（ ）。
- 旋转延迟的大小与磁盘调度算法无关
  - 旋转延迟的大小取决于磁盘空闲空间的分配程序
  - 旋转延迟的大小与文件的物理结构有关
  - 扇区数据的处理时间对旋转延迟的影响较大
8. 下列算法中，用于磁盘调度的是（ ）。
- 时间片轮转调度算法
  - LRU 算法
  - 最短寻找时间优先算法
  - 优先级高者优先算法
9. 以下算法中，（ ）可能出现“饥饿”现象。
- 电梯调度
  - 最短寻找时间优先
  - 循环扫描算法
  - 先来先服务
10. 在以下算法中，（ ）可能会随时改变磁头的运动方向。
- 电梯调度
  - 先来先服务
  - 循环扫描算法
  - 以上答案都不会
11. 已知某磁盘的平均转速为  $r$  秒/转，平均寻找时间为  $T$  秒，每个磁道可以存储的字节数为  $N$ ，现向该磁盘读写  $b$  字节的数据，采用随机寻道的方法，每道的所有扇区组成一个簇，其平均访问时间是（ ）。
- $(r + T)b/N$
  - $b/NT$
  - $(b/N + T)r$
  - $bT/N + r$
12. 设磁盘的转速为 3000 转/分，盘面划分为 10 个扇区，则读取一个扇区的时间为（ ）。
- 20ms
  - 5ms
  - 2ms
  - 1ms
13. 一个磁盘的转速为 7200 转/分，每个磁道有 160 个扇区，每扇区有 512B，那么理想情况下，其数据传输率为（ ）。
- $7200 \times 160 \text{KB/s}$
  - 7200KB/s
  - 9600KB/s
  - 19200KB/s
14. 【2009 统考真题】假设磁头当前位于第 105 道，正在向磁道序号增加的方向移动。现有一个磁道访问请求序列为 35, 45, 12, 68, 110, 180, 170, 195，采用 SCAN 调度（电梯调度）算法得到的磁道访问序列是（ ）。
- 110, 170, 180, 195, 68, 45, 35, 12
  - 110, 68, 45, 35, 12, 170, 180, 195
  - 110, 170, 180, 195, 12, 35, 45, 68
  - 12, 35, 45, 68, 110, 170, 180, 195
15. 设一个磁道访问请求序列为 55, 58, 39, 18, 90, 160, 150, 38, 184，磁头的起始位置为 100，若采用 SSTF（最短寻道时间优先）算法，则磁头移动（ ）个磁道。
- 55
  - 184
  - 200
  - 248
16. 假定磁带的记录密度为 400 字符/英寸 ( $1\text{in} = 0.0254\text{m}$ )，每条逻辑记录为 80 字符，块间隙为 0.4 英寸，现有 3000 个逻辑记录需要存储，存储这些记录需要长度为（ ）的磁带，磁带利用率是（ ）。
- 1500 英寸，33.3%
  - 1500 英寸，43.5%
  - 1800 英寸，33.3%
  - 1800 英寸，43.5%
17. 【2015 统考真题】某硬盘有 200 个磁道（最外侧磁道号为 0），磁道访问请求序列为 130, 42, 180, 15, 199，当前磁头位于第 58 号磁道并从外侧向内侧移动。按照 SCAN 调度方法处理完上述请求后，磁头移过的磁道数是（ ）。

- A. 208      B. 287      C. 325      D. 382
18. 【2017 统考真题】下列选项中，磁盘逻辑格式化程序所做的工作是（ ）。
- 对磁盘进行分区
  - 建立文件系统的根目录
  - 确定磁盘扇区校验码所占位数
  - 对保存空闲磁盘块信息的数据结构进行初始化
- A. 仅 II      B. 仅 II、IV      C. 仅 III、IV      D. 仅 I、II、IV
19. 【2018 统考真题】系统总是访问磁盘的某个磁道而不响应对其他磁道的访问请求，这种现象称为磁臂黏着。下列磁盘调度算法中，不会导致磁臂黏着的是（ ）。
- 先来先服务 (FCFS)
  - 最短寻道时间优先 (SSTF)
  - 扫描算法 (SCAN)
  - 循环扫描算法 (CSCAN)
20. 【2017 统考真题】某文件系统的簇和磁盘扇区大小分别为 1KB 和 512B。若一个文件的大小为 1026B，则系统分配给该文件的磁盘空间大小是（ ）。
- A. 1026B      B. 1536B      C. 1538B      D. 2048B

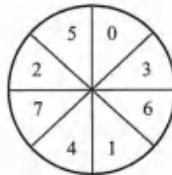
## 二、综合应用题

1. 假定有一个磁盘组共有 100 个柱面，每个柱面有 8 个磁道，每个磁道划分成 8 个扇区。现有一个 5000 条逻辑记录的文件，逻辑记录的大小与扇区大小相等，该文件以顺序结构被存放在磁盘组上，柱面、磁道、扇区均从 0 开始编址，逻辑记录的编号从 0 开始，文件信息从 0 柱面、0 磁道、0 扇区开始存放。试问，该文件编号为 3468 的逻辑记录应存放在哪个柱面的第几个磁道的第几个扇区上？
2. 【2010 统考真题】如下图所示，假设计算机系统采用 C-SCAN (循环扫描) 磁盘调度策略，使用 2KB 的内存空间记录 16384 个磁盘块的空闲状态。



- 1) 请说明在上述条件下如何进行磁盘块空闲状态的管理。
- 2) 设某单面磁盘旋转速度为 6000 转/分，每个磁道有 100 个扇区，相邻磁道间的平均移动时间为 1ms。若在某时刻，磁头位于 100 号磁道处，并沿着磁道号增大的方向移动（见上图），磁道号请求队列为 50, 90, 30, 120，对请求队列中的每个磁道需读取 1 个随机分布的扇区，则读完这 4 个扇区点共需要多少时间？要求给出计算过程。
- 3) 若将磁盘替换为随机访问的 Flash 半导体存储器（如 U 盘、固态硬盘等），是否有比 C-SCAN 更高效的磁盘调度策略？若有，给出磁盘调度策略的名称并说明理由；若无，说明理由。
3. 假设磁盘的每个磁道分成 9 个块，现在一个文件有 A, B, …, I 共 9 条记录，每条记录的大小与块的大小相等，设磁盘转速为 27ms/转，每读出一块后需要 2ms 的处理时间。若忽略

其他辅助时间，试问：

- 1) 若顺序存放这些记录顺序读取，处理该文件要多少时间？
- 2) 若要顺序读取该文件，记录如何存放处理时间最短？
4. 在一个磁盘上，有 1000 个柱面，编号为 0 ~ 999，用下面的算法计算为满足磁盘队列中的所有请求，磁盘臂必须移过的磁道的数目。假设最后服务的请求是在磁道 345 上，并且读写头正在朝磁道 0 移动。在按 FCFS 顺序排列的队列中包含了如下磁道上的请求：123, 874, 692, 475, 105, 376。
  - 1) FCFS; 2) SSTF; 3) SCAN; 4) LOOK; 5) C-SCAN; 6) C-LOOK。
5. 某软盘有 40 个磁道，磁头从一个磁道移至相邻磁道需要 6ms。文件在磁盘上非连续存放，逻辑上相邻数据块的平均距离为 13 磁道，每块的旋转延迟时间及传输时间为 100ms 和 25ms，问读取一个 100 块的文件需要多少时间？若系统对磁盘进行了整理，让同一文件的磁盘块尽可能靠拢，从而使逻辑上相邻数据块的平均距离降为 2 磁道，这时读取一个 100 块的文件需要多少时间？
6. 有一个交叉存放信息的磁盘，信息在其上的存放方法如右图所示。每个磁道有 8 个扇区，每个扇区 512B，旋转速度为 3000 转/分，顺时针读扇区。假定磁头已在读取信息的磁道上，0 扇区转到磁头下需要 1/2 转，且设备对应的控制器不能同时进行输入/输出，在数据从控制器传递至内存的这段时间内，从磁头下通过的扇区数为 2，问依次读出一个磁道上的所有扇区需要多少时间？其数据传输速率为多少？
 
7. 【2019 统考真题】某计算机系统中的磁盘有 300 个柱面，每个柱面有 10 个磁道，每个磁道有 200 个扇区，扇区大小为 512B。文件系统的每个簇包含 2 个扇区。请回答下列问题：
  - 1) 磁盘的容量是多少？
  - 2) 假设磁头在 85 号柱面上，此时有 4 个磁盘访问请求，簇号分别为 100260, 60005, 101660 和 110560。采用最短寻道时间优先 (SSTF) 调度算法，系统访问簇的先后次序是什么？
  - 3) 第 100530 簇在磁盘上的物理地址是什么？将簇号转换成磁盘物理地址的过程是由 I/O 系统的什么程序完成的？

### 4.3.6 答案与解析

#### 一、单项选择题

1. D

磁盘是可共享设备（分时共享），是指某段时间内可以有多个用户进行访问。但某一时刻只能有一个作业可以访问。

2. A

磁带是一种顺序存储设备，用它存储文件时只能采用顺序存储结构。注意：若允许磁带来回倒带，也可组织为其他的文件形式，本题不做讨论。

3. B

顺序访问按从前到后的顺序对数据进行读写操作，如磁带。随机访问，即直接访问，可以按任意的次序对数据进行读写操作，如光盘、磁盘、U 盘等。

4. A

磁盘调度是对访问磁道次序的调度，若没有合适的磁盘调度，则寻找时间会大大增加。

5. A

文件以块为单位存放于磁盘中，文件的读写也以块为单位。

#### 6. D

磁盘调度中，对读/写时间影响最大的是寻找时间，寻找过程为机械运动，时间较长，影响较大。

#### 7. D

磁盘调度算法是为了减少寻找时间。扇区数据的处理时间主要影响传输时间。选项 B、C 均与旋转延迟有关，文件的物理结构与磁盘空间的分配方式相对应，包括连续分配、链接分配和索引分配。连续分配的磁盘中，文件的物理地址连续；而链接分配方式的磁盘中，文件的物理地址不连续，因此与旋转延迟都有关。

#### 8. C

选项 A 是进程调度算法；选项 B 是页面淘汰算法；选项 D 可以用于进程调度和作业调度。只有选项 C 是磁盘调度算法。

#### 9. B

最短寻找时间优先算法中，当新的距离磁头比较近的磁盘访问请求不断被满足时，可能会导致较远的磁盘访问请求被无限延迟，从而导致“饥饿”现象。

#### 10. B

先来先服务算法根据磁盘请求的时间先后进行调度，因而可能随时改变磁头方向。而电梯调度、循环扫描算法均限制磁头的移动方向。

#### 11. A

将每道的所有扇区组成一个簇，意味着可以将一个磁道的所有存储空间组织成一个数据块组，这样有利于提高存储速度。读写磁盘时，磁头首先找到磁道，称为寻道，然后才可以将信息从磁道里读出或写入。读写完一个磁道后，磁头会继续寻找下一个磁道，完成剩余的工作，所以在随机寻道的情况下，读写一个磁道的时间要包括寻道时间和读写磁道时间，即  $T + r$  秒。由于总的数据量是  $b$  字节，它要占用的磁道数为  $b/N$  个，所以总平均读写时间为  $(r + T)b/N$  秒。

#### 12. C

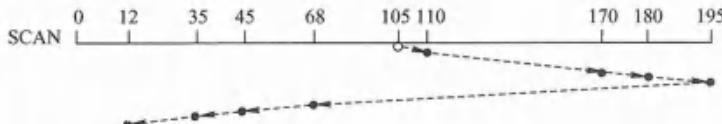
访问每条磁道的时间为  $60/3000s = 0.02s = 20ms$ ，即磁盘旋转一圈的时间为 20ms，每个盘面 10 个扇区，因此读取一个扇区的时间为  $20ms/10 = 2ms$ 。

#### 13. C

磁盘的转速为 7200 转/分 = 120 转/秒，转一圈经过 160 个扇区，每个扇区为 512B，所以数据传输率 =  $120 \times 160 \times 512 / 1024KB/s = 9600KB/s$ 。

#### 14. A

SCAN 算法的原理类似于电梯。首先，当磁头从 105 道向序号增加的方向移动时，便会按照从小到大的顺序服务所有大于 105 的磁道号（110, 170, 180, 195）；往回移动时又会按照从大到小的顺序进行服务（68, 45, 35, 12），结果如下图所示。



#### 15. D

对 SSTF 算法，寻道序列应为 100, 90, 58, 55, 39, 38, 18, 150, 160, 184；移动磁道次数分别为 10, 32, 3, 16, 1, 20, 132, 10, 24，总数为 248。

16. C

一个逻辑记录所占的磁带长度为  $80/400 = 0.2$  英寸，因此存储 3000 条逻辑记录需要的磁带长度为  $(0.2 + 0.4) \times 3000 = 1800$  英寸，利用率为  $0.2/(0.2 + 0.4) = 33.3\%$ 。

17. C

SCAN 算法就是电梯调度算法。顾名思义，若开始时磁头向外移动，就一直要到最外侧，然后返回向内侧移动，就像电梯若往下则一直要下到底层才会再上升一样。当前磁头位于 58 号并从外侧向内侧移动，先依次访问 130 和 199，然后返回向外侧移动，依次访问 42 和 15，因此磁头移过的磁道数是  $(199 - 58) + (199 - 15) = 325$ 。

18. B

新磁盘是空白的，必须分成各个扇区以便磁盘控制器能读和写，这个过程称为低级格式化（或物理格式化）。低级格式化为磁盘的每个扇区采用特别的数据结构，包括校验码，III 错误。为了使用磁盘存储文件，操作系统还需要将自己的数据结构记录在磁盘上。这分为两步。第一步是将磁盘分为由一个或多个柱面组成的分区，每个分区可以作为一个独立的磁盘，I 错误。在分区之后，第二步是逻辑格式化（创建文件系统）。在这一步，操作系统将初始的文件系统数据结构存储到磁盘上。这些数据结构包括空闲和已分配的空间及一个初始为空的目录，II、IV 正确。所以选 B。

19. A

当系统总是持续出现某个磁道的访问请求时，均持续满足最短寻道时间优先、扫描算法和循环扫描算法的访问条件，会一直服务该访问请求。而先来先服务按照请求次序进行调度，比较公平，因此选 A。

20. D

绝大多数操作系统为改善磁盘访问时间，以簇为单位进行空间分配，因此答案选 D。

## 二、综合应用题

1. 解答：

该磁盘有 8 个盘面，一个柱面大小为  $8 \times 8 = 64$  个扇区，即 64 条逻辑记录。由于所有磁头是固定在一起的，因此在存放数据时，先存满扇区，后存满磁道，再存满柱面。

编号为 3468 的逻辑记录对应的柱面号为  $3468/64 = 54$ ；对应的磁道号为  $(3468 \bmod 64) \text{ DIV } 8 = 1$ ；对应的扇区号为  $(3468 \bmod 64) \bmod 8 = 4$ 。

2. 解答：

1) 用位图表示磁盘的空闲状态。每位表示一个磁盘块的空闲状态，共需  $16 \times 384/32 = 512$  个字  $= 512 \times 4B = 2KB$ ，正好可放在系统提供的内存中。

2) 采用 C-SCAN 调度算法，访问磁道的顺序和移动的磁道数如下表所示：

被访问的下一个磁道号	移动距离（磁道数）
120	20
30	90
50	20
90	40

移动的磁道数为  $20 + 90 + 20 + 40 = 170$ ，因此总的移动磁道时间为 170ms。

由于转速为 6000 转/分，因此平均旋转延迟为 5ms，总的旋转延迟时间 = 20ms。

由于转速为 6000 转/分，因此读取一个磁道上的一个扇区的平均读取时间为 0.1ms，扇区

的平均读取时间为 0.1ms，总的读取扇区的时间为 0.4ms。

综上，读取上述磁道上所有扇区所花的总时间为 190.4ms。

- 3) 采用先来先服务（FCFS）调度策略更高效。因为 Flash 半导体存储器的物理结构不需要考虑寻道时间和旋转延迟，可直接按 I/O 请求的先后顺序服务。

### 3. 解答：

由题目所给条件可知，磁盘转速为 27ms/转，每个磁道存放 9 条记录，因此读出 1 条记录的时间是  $27/9 = 3\text{ms}$ 。

- 1) 读出并处理记录 A 需要 5ms，此时读写头已转到记录 B 的中间，因此为了读出记录 B，必须再转接近一圈（从记录 B 的中间到记录 B）。后续 8 条记录的读取及处理与此相同，但最后一条记录的读取与处理只需要 5ms。于是，处理 9 条记录的总时间为

$$8 \times (27 + 3) + (3 + 2) = 245\text{ms}$$

- 2) 由于读出并处理一条记录需要 5ms，当读出并处理记录 A 时，不妨设记录 A 放在第 1 个盘块中，读写头已移到第 2 个盘块的中间，为了能顺序读到记录 B，应将它放到第 3 个盘块中，即应将记录按下表顺序存放：

盘块	1	2	3	4	5	6	7	8	9
记录	A	F	B	G	C	H	D	I	E

这样，处理一条记录并将磁头移到下一条记录的时间是

$$3 \text{ (读出)} + 2 \text{ (处理)} + 1 \text{ (等待)} = 6\text{ms}$$

所以，处理 9 条记录的总时间为

$$6 \times 8 + 5 = 53\text{ms}$$

### 4. 解答：

- 1) FCFS：移动磁道的顺序为 345, 123, 874, 692, 475, 105, 376。磁盘臂必须移过的磁道的数目为  $222 + 751 + 182 + 217 + 370 + 271 = 2013$ 。

- 2) SSTF：移动磁道的顺序为 345, 376, 475, 692, 874, 123, 105。磁盘臂必须移过的磁道的数目为  $31 + 99 + 217 + 182 + 751 + 18 = 1298$ 。

注意：磁盘臂必须移过的磁道的数目之和的计算没有必要像上面一样对 31, 99, 217, 182, 751, 18 求和，仔细的读者会发现：从 345 到 874 是一路递增的，接着从 874 到 105 是一路递减的。所以仅需计算  $(874 - 345) + (874 - 105) = 1298$ 。这种方法是不是要比上面得出 6 个数后再计算它们的和要快捷一些？若之前未注意到此法，相信聪明的读者会马上回顾刚做完的 1)，并会仔细观察以下几问的“规律”，进而总结出自己的思路。

- 3) SCAN：移动磁道的顺序为 345, 123, 105, 0, 376, 475, 692, 874。磁盘臂必须移过的磁道的数目为  $222 + 18 + 105 + 376 + 99 + 217 + 182 = 1219$ 。

- 4) LOOK：移动磁道的顺序为 345, 123, 105, 376, 475, 692, 874。磁盘臂必须移过的磁道的数目为  $222 + 18 + 271 + 99 + 217 + 182 = 1009$ 。

- 5) C-SCAN：移动磁道的顺序为 345, 123, 105, 0, 999, 874, 692, 475, 376。磁盘臂必须移过的磁道的数目为  $222 + 18 + 105 + 999 + 125 + 182 + 217 + 99 = 1967$ 。

- 6) C-LOOK：移动磁道的顺序为 345, 123, 105, 874, 692, 475, 376。磁盘臂必须移过的磁道的数目为  $222 + 18 + 769 + 182 + 217 + 99 = 1507$ 。

### 5. 解答：

磁盘整理前，逻辑上相邻数据块的平均距离为 13 磁道，读一块数据需要的时间为

$$13 \times 6 + 100 + 25 = 203\text{ms}$$

因此，读取一个 100 块的文件需要的时间为

$$203 \times 100 = 20300\text{ms}$$

磁盘整理后，逻辑上相邻数据块的平均距离为 2 磁道，读一块数据需要的时间为

$$2 \times 6 + 100 + 25 = 137\text{ms}$$

因此，读取一个 100 块的文件需要的时间为

$$137 \times 100 = 13700\text{ms}$$

#### 6. 解答：

从图中可知信息块之间的间隔为 2 个扇区。由题目所给的条件可知，旋转速度为 3000 转/分 = 50 转/秒，即 20ms/转。

读一个扇区需要的时间为

$$20/8 = 2.5\text{ms}$$

读一个扇区并将扇区数据送入内存需要的时间为

$$2.5 \times 3 = 7.5\text{ms}$$

读出一个磁道上的所有扇区需要的时间为

$$20/2 + 8 \times 7.5 = 70\text{ms} = 0.07\text{s}$$

每个磁道的数据量为

$$8 \times 512 = 4\text{KB}$$

数据传输速率为

$$4\text{KB}/0.07\text{s} = 4 \times 1024\text{B}/(1000 \times 0.07\text{s}) = 58.5\text{kB/s}$$

所以，依次读出一个磁道上的所有扇区需要 0.07s，其数据传输速率为 58.5kB/s。

注意：表示存储容量、文件大小时 K 等于 1024，表示传输速率时 K 等于 1000，注意转换。

#### 7. 解答：

1) 磁盘容量 = 磁盘的柱面数×每个柱面的磁道数×每个磁道的扇区数×每个扇区的大小 =  $(300 \times 10 \times 200 \times 512/1024)\text{KB} = 3 \times 10^5\text{KB}$ 。

2) 磁头在 85 号柱面上，对 SSTF 算法而言，总是访问当前柱面距离最近的地址。注意每个簇包含 2 个扇区，通过计算得到，85 号柱面对应的簇号为 85000~85999。通过比较得出，系统最先访问离 85000~85999 最近的 100260，随后访问离 100260 最近的 101660，然后访问 110560，最后访问 60005。顺序为 100260, 101660, 110560, 60005。

3) 第 100530 簇在磁盘上的物理地址由其所在的柱面号、磁头号、扇区号构成。

$$\text{柱面号} = \lfloor \text{簇号}/(\text{每个柱面的簇数}) \rfloor = \lfloor 100530/(10 \times 200/2) \rfloor = 100.$$

$$\text{磁头号} = \lfloor (\text{簇号} \% \text{每个柱面的簇数}) / \text{每个磁道的簇数} \rfloor = \lfloor 530/(200/2) \rfloor = 5.$$

$$\text{扇区号} = \text{扇区地址} \% \text{每个磁道的扇区数} = (530 \times 2) \% 200 = 60.$$

将簇号转换成磁盘物理地址的过程由磁盘驱动程序完成。

## 4.4 本章疑难点

### 1. 磁盘结构

引导控制块（Boot Control Block）包括系统从该分区引导操作系统所需要的信息。若磁盘没

有操作系统，则这块的内容为空。它通常为分区的第一块。UFS 称为引导块（Boot Block）；NTFS 称为分区引导扇区（Partition Boot Sector）。

分区控制块（Partition Control Block）包括分区详细信息，如分区的块数、块的大小、空闲块的数量和指针、空闲 FCB 的数量和指针等。UFS 称为超级块（Super Block），而 NTFS 称为主控文件表（Master File Table）。

## 2. 内存结构

内存分区表包含所有安装分区的信息。

内存目录结构用来保存近来访问过的目录信息。对安装分区的目录，可以包括一个指向分区表的指针。

系统范围的打开文件表，包括每个打开文件的 FCB 复制和其他信息。

单个进程的打开文件表，包括一个指向系统范围内已打开文件表中合适条目和其他信息的指针。

## 3. 文件系统实现概述

为了创建一个文件，应用程序调用逻辑文件系统。逻辑文件系统知道目录结构形式，它将为文件分配一个新的 FCB，把相应目录读入内存，用新的文件名更新该目录和 FCB，并将结果写回磁盘。图 4.25 显示了一个典型的 FCB。

一旦文件被创建，它就能用于 I/O，不过首先要打开文件。调用 open 将文件名传给文件系统，文件系统根据给定文件名搜索目录结构。部分目录结构通常缓存在内存中以加快目录操作。找到文件后，其 FCB 复制到系统范围的打开文件表。该表不但存储 FCB，而且存储打开该文件的进程数量的条目。

然后，单个进程的打开文件表中会增加一个条目，并通过指针将系统范围的打开文件表的条目与其他域（文件当前位置的指针和文件打开模式等）相连。调用 open 返回的是一个指向单个进程的打开文件表中合适条目的指针，所以文件操作都是通过该指针进行的。

文件名不必是打开文件表的一部分，因为一旦完成对 FCB 在磁盘上的定位，系统就不再使用文件名。对于访问打开文件表的索引，UNIX 称之为文件描述符（File Descriptor），而 Windows 2000 称之为文件句柄（File Handle）。因此，只要文件未被关闭，所有文件操作就通过打开文件表来进行。

当一个进程关闭文件时，就会删除单个进程打开文件表中的一个相应条目，即目录项，系统范围内打开文件表的打开数也会递减。当打开文件的所有用户都关闭了一个文件时，更新的文件信息会复制到磁盘的目录结构中，系统范围的打开文件表的条目也将删除。

在实际中，系统调用 open 时会首先搜索系统范围的打开文件表，以确定某文件是否已被其他进程所使用。如果是，就在单个进程的打开文件表中创建一项，并指向现有系统范围的打开文件表的相应条目。该算法在文件已打开时，能节省大量开销。

## 4. 混合索引分配的实现

混合索引分配已在 UNIX 系统中采用。在 UNIX System V 的索引结点中，共设置了 13 个地址项，即 iaddr(0)~iaddr(12)，如图 4.26 所示。在 BSD UNIX 的索引结点中，共设置了 13 个地址项，它们都把所有的地址项分成两类，即直接地址和间接地址。

文件权限
文件日期（创建，访问，写）
文件所有者，组，ACL
文件大小
文件数据块

图 4.25 典型的 FCB

### (1) 直接地址

为了提高对文件的检索速度，在索引结点中可设置 10 个直接地址项，即用  $iaddr(0) \sim iaddr(9)$  来存放直接地址。换言之，这里每项中所存放的是该文件数据所在盘块的盘块号。假如每个盘块的大小为 4KB，当文件不大于 40KB 时，便可直接从索引结点中读出该文件的全部盘块号。

### (2) 一次间接地址

对于大、中型文件，只采用直接地址并不现实。可再利用索引结点中的地址项  $iaddr(10)$  来提供一次间接地址。这种方式的实质就是一级索引分配方式。图中的一次间址块也就是索引块，系统将分配给文件的多个盘块号记入其中。在一次间址块中可存放 1024 个盘块号，因而允许文件长达 4MB。

### (3) 多次间接地址

当文件长度大于  $4MB + 40KB$  (一次间接地址与 10 个直接地址项) 时，系统还须采用二次间接地址分配方式。这时，用地址项  $iaddr(11)$  提供二次间接地址。该方式的实质是两级索引分配方式。系统此时在二次间接地址块中记入所有一次间接地址块的盘号。在采用二次间接地址方式时，文件的最大长度可达 4GB。同理，地址项  $iaddr(12)$  作为三次间接地址，其允许的文件最大长度可达 4TB。

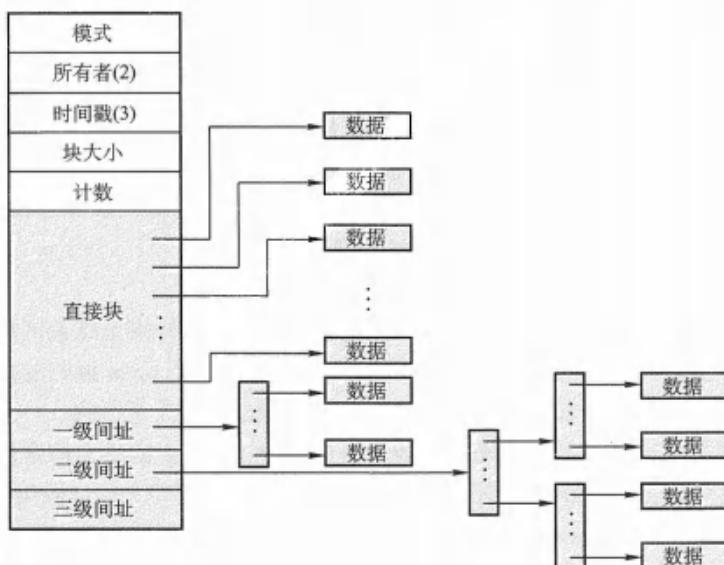


图 4.26 UNIX 系统的 inode 结构示意图

最后，我们对本章内容再进行一次宏观上的把握。贯穿本章内容的有两条主线：第一条主线是介绍一种新的抽象数据类型、文件，从逻辑结构和物理结构两个方面进行；第二条主线是操作系统是如何管理“文件”这种数据结构的，介绍了多文件的逻辑结构的组织，即目录，还介绍了如何处理用户对文件的服务请求，即磁盘管理。但宏观认识是远远不够的，从宏观上把握知识的目的是从微观上更加准确地掌控细微知识点，在考试中得到好成绩。读者要通过反复做题、对答案，不断加深自己对知识点的认知程度。

# 第5章

## 输入/输出 (I/O) 管理

### 【考纲内容】

#### (一) I/O 管理概述

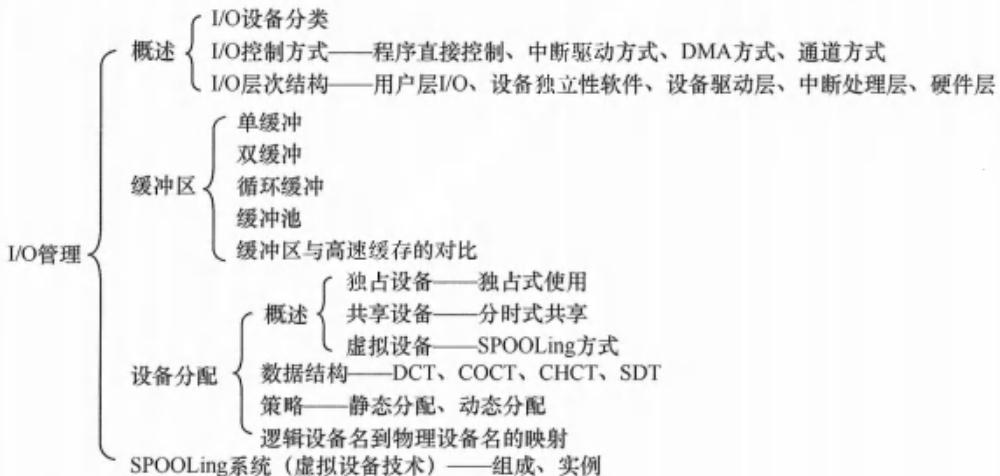
I/O 控制方式; I/O 软件层次结构

#### (二) I/O 核心子系统

I/O 调度概念; 高速缓存与缓冲区

设备分配与回收; 假脱机技术 (SPOOLing)

### 【知识框架】



### 【复习提示】

本章的内容较为分散，重点掌握的内容是 I/O 设备的基本特性、I/O 子系统的特性、三种 I/O 控制方式、高速缓存与缓冲区、SPOOLing 技术。本章的知识点很多，如 I/O 方式、设备控制等内容与硬件直接相关，建议结合计算机组成原理中的对应章节一起复习。本章内容与组成原理中的交叉较多，很多考点既可作为本章的考点，又可作为组成原理中的考点，因此还未复习过组成原理的读者需要清楚地把握本章的每个知识点，为组成原理的学习打下基础，已复习过组成原理的读者遇到比较熟悉的内容时可以跳过，学习本章中组成原理未涉及的部分即可。另外，未复习过组成原理的读者可能会觉得本章的习题较难，但不需要担心。

本章的内容历年来在统考题目中所占的比例不大，若统考中出现本章的题目，则基本上可以断定一定非常简单，看过相关内容的读者就一定会做，而未看过的读者基本上只能靠“蒙”。考研成功的秘诀是复习要反复多次并全面，偷工减料是要吃亏的，希望读者重视本章的内容。

## 5.1 I/O 管理概述

学习本章时，可与计算机组成原理的相关知识相结合，并思考 I/O 管理要完成哪些功能。

### 5.1.1 I/O 设备

I/O 设备管理是操作系统设计中最凌乱也最具挑战性的部分。由于它包含了很多领域的不同设备及与设备相关的应用程序，因此很难有一个通用且一致的设计方案。所以在理解设备管理之前，应该先了解具体的 I/O 设备类型。

计算机系统中的 I/O 设备按使用特性可分为以下类型：

- 1) 人机交互类外部设备。用于与计算机用户之间交互的设备，如打印机、显示器、鼠标、键盘等。这类设备的数据交换速度相对较慢，通常是以字节为单位进行数据交换的。
- 2) 存储设备。用于存储程序和数据的设备，如磁盘、磁带、光盘等。这类设备用于数据交换，速度较快，通常以多字节组成的块为单位进行数据交换。
- 3) 网络通信设备。用于与远程设备通信的设备，如各种网络接口、调制解调器等。其速度介于前两类设备之间。网络通信设备在使用和管理上与前两类设备也有很大不同。

除了上面最常见的分类方法，I/O 设备还可以按以下方法分类。

#### 1. 按传输速率分类

- 1) 低速设备。传输速率仅为每秒几字节到数百字节的一类设备，如键盘、鼠标等。
- 2) 中速设备。传输速率为每秒数千字节至数万字节的一类设备，如行式打印机、激光打印机等。
- 3) 高速设备。传输速率在数百千字节至千兆字节的一类设备，如磁带机、磁盘机、光盘机等。

#### 2. 按信息交换的单位分类

- 1) 块设备。由于信息的存取总是以数据块为单位的，所以存储信息的设备称为块设备。它属于有结构设备，如磁盘等。磁盘设备的基本特征是传输速率较高、可寻址，即对它可随机地读/写任一块。
- 2) 字符设备。用于数据输入/输出的设备为字符设备，因为其传输的基本单位是字符。它属于无结构类型，如交互式终端机、打印机等。它们的基本特征是传输速率低、不可寻址，并且在输入/输出时常采用中断驱动方式。

### 5.1.2 I/O 控制方式<sup>①</sup>

设备管理的主要任务之一是控制设备和内存或处理机之间的数据传送。外围设备和内存之间的输入/输出控制方式有 4 种，下面分别加以介绍。

#### 1. 程序直接控制方式

如图 5.1(a)所示，计算机从外部设备读取数据到存储器，每次读一个字的数据。对读入的每个字，CPU 需要对外设状态进行循环检查，直到确定该字已经在 I/O 控制器的数据寄存器中。在

<sup>①</sup> 建议结合《计算机组成原理统考复习指导》第 7 章的内容进行学习。

程序直接控制方式中,由于CPU的高速性和I/O设备的低速性,致使CPU的绝大部分时间都处于等待I/O设备完成数据I/O的循环测试中,造成了CPU资源的极大浪费。在该方式中,CPU之所以要不断地测试I/O设备的状态,就是在CPU中未采用中断机构,使I/O设备无法向CPU报告它已完成了字符的输入操作。

程序直接控制方式虽然简单且易于实现,但其缺点也显而易见,由于CPU和I/O设备只能串行工作,导致CPU的利用率相当低。

## 2. 中断驱动方式

中断驱动方式的思想是,允许I/O设备主动打断CPU的运行并请求服务,从而“解放”CPU,使得其向I/O控制器发送读命令后可以继续做其他有用的工作。如图5.1(b)所示,我们从I/O控制器和CPU两个角度分别来看中断驱动方式的工作过程。

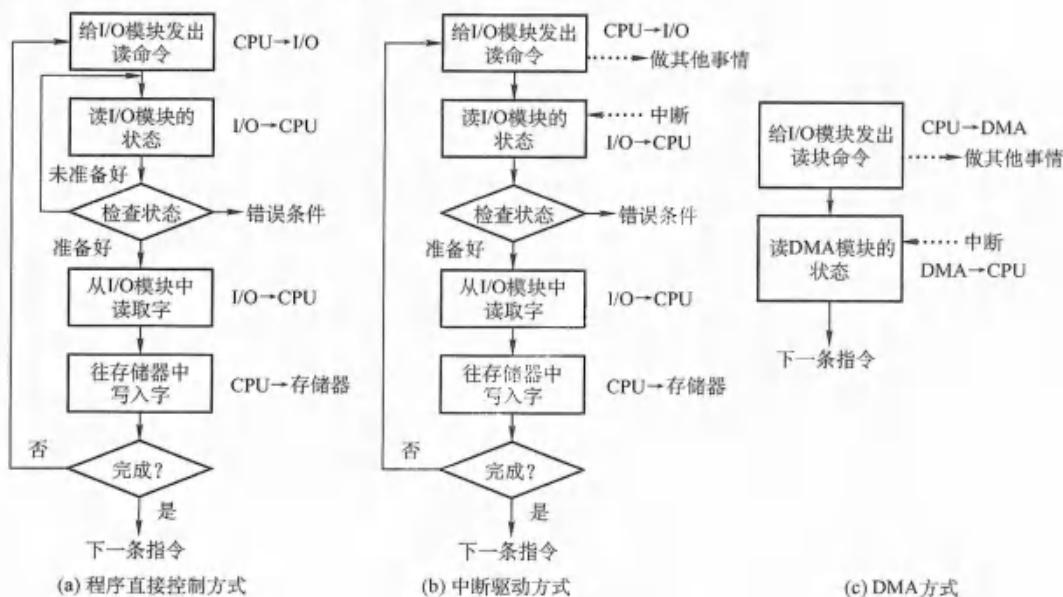


图5.1 I/O控制方式

从I/O控制器的角度来看,I/O控制器从CPU接收一个读命令,然后从外围设备读数据。一旦数据读入该I/O控制器的数据寄存器,便通过控制线给CPU发出一个中断信号,表示数据已准备好,然后等待CPU请求该数据。I/O控制器收到CPU发出的取数据请求后,将数据放到数据总线上,传到CPU的寄存器中。至此,本次I/O操作完成,I/O控制器又可开始下一次I/O操作。

从CPU的角度来看,CPU发出读命令,然后保存当前运行程序的上下文(现场,包括程序计数器及处理机寄存器),转去执行其他程序。在每个指令周期的末尾,CPU检查中断。当有来自I/O控制器的中断时,CPU保存当前正在运行程序的上下文,转去执行中断处理程序以处理该中断。这时,CPU从I/O控制器读一个字的数据传送到寄存器,并存入主存。接着,CPU恢复发出I/O命令的程序(或其他程序)的上下文,然后继续运行。

中断驱动方式比程序直接控制方式有效,但由于数据中的每个字在存储器与I/O控制器之间的传输都必须经过CPU,这就导致了中断驱动方式仍然会消耗较多的CPU时间。

## 3. DMA方式

在中断驱动方式中,I/O设备与内存之间的数据交换必须要经过CPU中的寄存器,所以速度

还是受限，而 DMA（直接存储器存取）方式的基本思想是在 I/O 设备和内存之间开辟直接的数据交换通路，彻底“解放”CPU。DMA 方式的特点如下：

- 1) 基本单位是数据块。
- 2) 所传送的数据，是从设备直接送入内存的，或者相反。
- 3) 仅在传送一个或多个数据块的开始和结束时，才需 CPU 干预，整块数据的传送是在 DMA 控制器的控制下完成的。

图 5.2 列出了 DMA 控制器的组成。

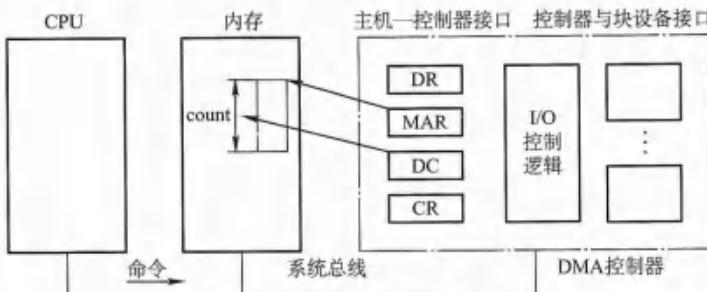


图 5.2 DMA 控制器的组成

要在主机与控制器之间实现成块数据的直接交换，须在 DMA 控制器中设置如下 4 类寄存器：

- 1) 命令/状态寄存器 (CR)。用于接收从 CPU 发来的 I/O 命令或有关控制信息，或设备的状态。
- 2) 内存地址寄存器 (MAR)。在输入时，它存放把数据从设备传送到内存的起始目标地址；在输出时，它存放由内存到设备的内存源地址。
- 3) 数据寄存器 (DR)。用于暂存从设备到内存或从内存到设备的数据。
- 4) 数据计数器 (DC)。存放本次要传送的字 (节) 数。

如图 5.1(c)所示，DMA 方式的工作过程是：CPU 接收到 I/O 设备的 DMA 请求时，它给 I/O 控制器发出一条命令，启动 DMA 控制器，然后继续其他工作。之后 CPU 就把控制操作委托给 DMA 控制器，由该控制器负责处理。DMA 控制器直接与存储器交互，传送整个数据块，每次传送一个字，这个过程不需要 CPU 参与。传送完成后，DMA 控制器发送一个中断信号给处理器。因此只有在传送开始和结束时才需要 CPU 的参与。

DMA 控制方式与中断驱动方式的主要区别是，中断驱动方式在每个数据需要传输时中断 CPU，而 DMA 控制方式则是在所要求传送的一批数据全部传送结束时才中断 CPU；此外，中断驱动方式数据传送是在中断处理时由 CPU 控制完成的，而 DMA 控制方式则是在 DMA 控制器的控制下完成的。

#### 4. 通道控制方式

I/O 通道是指专门负责输入/输出的处理机。I/O 通道方式是 DMA 方式的发展，它可以进一步减少 CPU 的干预，即把对一个数据块的读（或写）为单位的干预，减少为对一组数据块的读（或写）及有关控制和管理为单位的干预。同时，又可以实现 CPU、通道和 I/O 设备三者的并行操作，从而更有效地提高整个系统的资源利用率。

例如，当 CPU 要完成一组相关的读（或写）操作及有关控制时，只需向 I/O 通道发送一条 I/O 指令，以给出其所要执行的通道程序的首地址和要访问的 I/O 设备，通道接到该指令后，执行通道程序便可完成 CPU 指定的 I/O 任务，数据传送结束时向 CPU 发中断请求。

I/O 通道与一般处理机的区别是：通道指令的类型单一，没有自己的内存，通道所执行的通道程序是放在主机的内存中的，也就是说通道与 CPU 共享内存。

I/O 通道与 DMA 方式的区别是：DMA 方式需要 CPU 来控制传输的数据块大小、传输的内存位置，而通道方式中这些信息是由通道控制的。另外，每个 DMA 控制器对应一台设备与内存传递数据，而一个通道可以控制多台设备与内存的数据交换。

下面用一个例子来总结以上 4 种 I/O 控制方式。想象一位客户要去裁缝店做一批衣服的情形。

采用程序直接控制时，裁缝没有客户的联系方式，客户必须每隔一段时间去裁缝店看看裁缝把衣服做好了没有，这就浪费了客户不少的时间。

采用中断驱动方式时，裁缝有客户的联系方式，每当他完成一件衣服后，给客户打一个电话，让客户去拿，与程序直接控制能省去客户不少麻烦，但每完成一件衣服就让客户去拿一次，仍然比较浪费客户的时间。

采用 DMA 方式时，客户花钱雇一位单线秘书，并向秘书交代好把衣服放在哪里（存放仓库），裁缝要联系就直接联系秘书，秘书负责把衣服取回来并放在合适的位置，每处理完 100 件衣服，秘书就要给客户报告一次（大大节省了客户的时间）。

采用通道方式时，秘书拥有更高的自主权，与 DMA 方式相比，他可以决定把衣服存放在哪里，而不需要客户操心。而且，何时向客户报告，是处理完 100 件衣服就报告，还是处理完 10000 件衣服才报告，秘书是可以决定的。客户有可能在多个裁缝那里订了货，一位 DMA 类的秘书只能负责与一位裁缝沟通，但通道类秘书却可以与多名裁缝进行沟通。

### 5.1.3 I/O 子系统的层次结构

I/O 软件涉及的面非常广，往下与硬件有着密切的联系，往上又与用户直接交互，它与进程管理、存储器管理、文件管理等都存在着一定的联系，即它们都可能需要 I/O 软件来实现 I/O 操作。

为了使复杂的 I/O 软件具有清晰的结构、良好的可移植性和适应性，在 I/O 软件中普遍采用了层次式结构，将系统输入/输出功能组织成一系列的层次，每层都利用其下层提供的服务，完成输入/输出功能中的某些子功能，并屏蔽这些功能实现的细节，向高层提供服务。在层次式结构的 I/O 软件中，只要层次间的接口不变，对某一层次中的软件的修改都不会引起其下层或高层代码的变更，仅最低层才涉及硬件的具体特性。

一个比较合理的层次划分如图 5.3 所示。整个 I/O 系统可以视为具有 4 个层次的系统结构，各层次及其功能如下：

- 1) 用户层 I/O 软件。实现与用户交互的接口，用户可直接调用在用户层提供的、与 I/O 操作有关的库函数，对设备进行操作。

一般而言，大部分的 I/O 软件都在操作系统内部，但仍有一小部分在用户层，包括与用户程序链接在一起的库函数，以及完全运行于内核之外的一些程序。用户层软件必须通过一组系统调用来获取操作系统服务。

- 2) 设备独立性软件。用于实现用户程序与设备驱动器的统一接口、设备命令、设备保护及设备分配与释放等，同时为设备管理和数据传送提供必要的存储空间。

设备独立性也称设备无关性，使得应用程序独立于具体使用的物理设备。为实现设备独立性而引入了逻辑设备和物理设备这两个概念。在应用程序中，使用逻辑设备名来请求

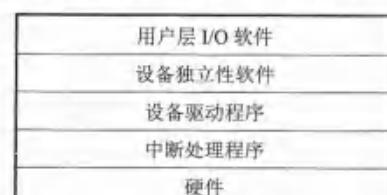


图 5.3 I/O 层次结构

使用某类设备；而在系统实际执行时，必须将逻辑设备名映射成物理设备名使用。

使用逻辑设备名的好处是：①增加设备分配的灵活性；②易于实现 I/O 重定向，所谓 I/O 重定向，是指用于 I/O 操作的设备可以更换（即重定向），而不必改变应用程序。

为了实现设备独立性，必须再在驱动程序之上设置一层设备独立性软件。总体而言，设备独立性软件的主要功能可分为以下两个方面：

① 执行所有设备的公有操作。包括：对设备的分配与回收；将逻辑设备名映射为物理设备名；对设备进行保护，禁止用户直接访问设备；缓冲管理；差错控制；提供独立于设备的大小统一的逻辑块，屏蔽设备之间信息交换单位大小和传输速率的差异。

② 向用户层（或文件层）提供统一接口。无论何种设备，它们向用户所提供的接口应是相同的。例如，对各种设备的读/写操作，在应用程序中都统一使用 read/write 命令等。

3) 设备驱动程序。与硬件直接相关，负责具体实现系统对设备发出的操作指令，驱动 I/O 设备工作的驱动程序。

通常，每类设备配置一个设备驱动程序，它是 I/O 进程与设备控制器之间的通信程序，常以进程形式存在。设备驱动程序向上层用户程序提供一组标准接口，设备具体的差别被设备驱动程序所封装，用于接收上层软件发来的抽象 I/O 要求，如 read 和 write 命令，转换为具体要求后，发送给设备控制器，控制 I/O 设备工作；它也将由设备控制器发来的信号传送给上层软件，从而为 I/O 内核子系统隐藏设备控制器之间的差异。

4) 中断处理程序。用于保存被中断进程的 CPU 环境，转入相应的中断处理程序进行处理，处理完并恢复被中断进程的现场后，返回到被中断进程。

中断处理层的主要任务有：进行进程上下文的切换，对处理中断信号源进行测试，读取设备状态和修改进程状态等。由于中断处理与硬件紧密相关，对用户而言，应尽量加以屏蔽，因此应放在操作系统的底层，系统的其余部分尽可能少地与之发生联系。

5) 硬件设备。I/O 设备通常包括一个机械部件和一个电子部件。为了达到设计的模块性和通用性，一般将其分开：电子部件称为设备控制器（或适配器），在个人计算机中，通常是一块插入主板扩充槽的印制电路板；机械部件则是设备本身。

设备控制器通过寄存器与 CPU 通信，在某些计算机上，这些寄存器占用内存地址的一部分，称为内存映像 I/O；另一些计算机则采用 I/O 专用地址，寄存器独立编址。操作系统通过向控制器寄存器写命令字来执行 I/O 功能。控制器收到一条命令后，CPU 可以转向进行其他工作，而让设备控制器自行完成具体的 I/O 操作。当命令执行完毕后，控制器发出一个中断信号，操作系统重新获得 CPU 的控制权并检查执行结果，此时，CPU 仍旧从控制器寄存器中读取信息来获得执行结果和设备的状态信息。

设备控制器的主要功能如下：

- 1) 接收和识别 CPU 或通道发来的命令，如磁盘控制器能接收读、写、查找等命令。
- 2) 实现数据交换，包括设备和控制器之间的数据传输；通过数据总线或通道，控制器和主存之间的数据传输。
- 3) 发现和记录设备及自身的状态信息，供 CPU 处理使用。
- 4) 设备地址识别。

为实现上述功能，设备控制器（见图 5.4）必须包含以下组成部分：

- ① 设备控制器与 CPU 的接口。该接口有三类信号线：数据线、地址线和控制线。数据线通常与两类寄存器相连：数据寄存器（存放从设备送来的输入数据或从 CPU 送来的输出数据）和控制/状态寄存器（存放从 CPU 送来的控制信息或设备的状态信息）。

- ② 设备控制器与设备的接口。设备控制器连接设备需要相应数量的接口，一个接口连接一台设备。每个接口中都存在数据、控制和状态三种类型的信号。
- ③ I/O 控制逻辑。用于实现对设备的控制。它通过一组控制线与 CPU 交互，对从 CPU 收到的 I/O 命令进行译码。CPU 启动设备时，将启动命令发送给控制器，同时通过地址线把地址发送给控制器，由控制器的 I/O 逻辑对地址进行译码，并相应地对所选设备进行控制。

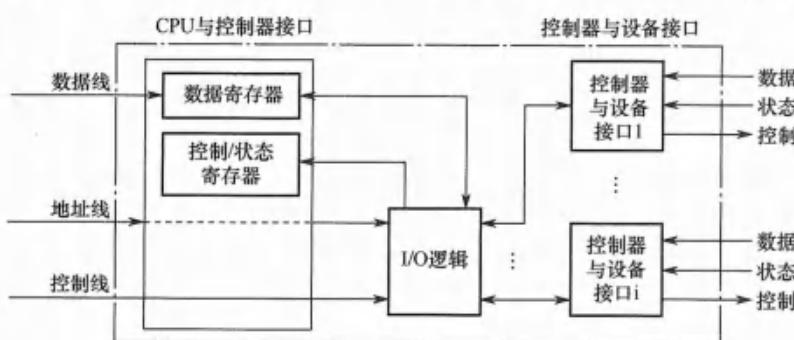


图 5.4 设备控制器的组成

类似于文件系统的层次结构，I/O 子系统的层次结构也是我们需要记忆的内容，但记忆不是死记硬背，我们以用户对设备的一次命令来总结各层次的功能，帮助各位读者记忆。

例如，当用户要读取某设备的内容时，通过操作系统提供的 read 命令接口，这就经过了用户层。

操作系统提供给用户使用的接口，一般是统一的通用接口，也就是几乎每个设备都可以响应的统一命令，如 read 命令，用户发出的 read 命令，首先经过设备独立层进行解析，然后往下层。

接下来，不同类型的设备对 read 命令的行为会有所不同，如磁盘接收 read 命令后的行为与打印机接收 read 命令后的行为是不同的。因此，需要针对不同的设备，把 read 命令解析成不同的指令，这就经过了设备驱动层。

命令解析完毕后，需要中断正在运行的进程，转而执行 read 命令，这就需要中断处理程序。

最后，命令真正抵达硬件设备，硬件设备的控制器按照上层传达的命令操控硬件设备，完成相应的功能。

#### 5.1.4 本节小结

本节开头提出的问题的参考答案如下。

I/O 管理要完成哪些功能？

I/O 管理需要完成以下 4 部分内容：

- 1) 状态跟踪。要能实时掌握外部设备的状态。
- 2) 设备存取。要实现对设备的存取操作。
- 3) 设备分配。在多用户环境下，负责设备的分配与回收。
- 4) 设备控制。包括设备的驱动、完成和故障的中断处理。

#### 5.1.5 本节习题精选

##### 一、单项选择题

1. 以下关于设备属性的叙述中，正确的是（ ）。

- A. 字符设备的基本特征是可寻址到字节，即能指定输入的源地址或输出的目标地址  
 B. 共享设备必须是可寻址的和可随机访问的设备  
 C. 共享设备是指同一时间内允许多个进程同时访问的设备  
 D. 在分配共享设备和独占设备时都可能引起进程死锁
2. 虚拟设备是指( )。  
 A. 允许用户使用比系统中具有的物理设备更多的设备  
 B. 允许用户以标准化方式来使用物理设备  
 C. 把一个物理设备变成多个对应的逻辑设备  
 D. 允许用户程序不必全部装入主存便可使用系统中的设备
3. 磁盘设备的 I/O 控制主要采取( )方式。  
 A. 位                    B. 字节                    C. 帧                    D. DMA
4. 为了便于上层软件的编制，设备控制器通常需要提供( )。  
 A. 控制寄存器、状态寄存器和控制命令  
 B. I/O 地址寄存器、工作方式状态寄存器和控制命令  
 C. 中断寄存器、控制寄存器和控制命令  
 D. 控制寄存器、编程空间和控制逻辑寄存器
5. 在设备控制器中用于实现设备控制功能的是( )。  
 A. CPU                    B. 设备控制器与处理器的接口  
 C. I/O 逻辑                    D. 设备控制器与设备的接口
6. 在设备管理中，设备映射表(DMT)的作用是( )。  
 A. 管理物理设备                    B. 管理逻辑设备  
 C. 实现输入/输出                    D. 建立逻辑设备与物理设备的对应关系
7. DMA 方式是在( )之间建立一条直接数据通路。  
 A. I/O 设备和主存                    B. 两个 I/O 设备  
 C. I/O 设备和 CPU                    D. CPU 和主存
8. 通道又称 I/O 处理机，它用于实现( )之间的信息传输。  
 A. 内存与外设                    B. CPU 与外设                    C. 内存与外存                    D. CPU 与外存
9. 在操作系统中，( )指的是一种硬件机制。  
 A. 通道技术                    B. 缓冲池                    C. SPOOLing 技术                    D. 内存覆盖技术
10. 若 I/O 设备与存储设备进行数据交换不经过 CPU 来完成，则这种数据交换方式是( )。  
 A. 程序查询                    B. 中断方式  
 C. DMA 方式                    D. 无条件存取方式
11. 计算机系统中，不属于 DMA 控制器的是( )。  
 A. 命令/状态寄存器                    B. 内存地址寄存器  
 C. 数据寄存器                    D. 堆栈指针寄存器
12. ( )用作连接大量的低速或中速 I/O 设备。  
 A. 数据选择通道                    B. 字节多路通道                    C. 数据多路通道                    D. I/O 处理机
13. 在下列问题中，( )不是设备分配中应考虑的问题。  
 A. 及时性                    B. 设备的固有属性                    C. 设备独立性                    D. 安全性
14. 将系统中的每台设备按某种原则统一进行编号，这些编号作为区分硬件和识别设备的代号，该编号称为设备的( )。

- A. 绝对号      B. 相对号      C. 类型号      D. 符号
15. 关于通道、设备控制器和设备之间的关系，以下叙述中正确的是（ ）。
- A. 设备控制器和通道可以分别控制设备  
 B. 对于同一组输入/输出命令，设备控制器、通道和设备可以并行工作  
 C. 通道控制设备控制器、设备控制器控制设备工作  
 D. 以上答案都不对
16. 有关设备管理的叙述中，不正确的是（ ）。
- A. 通道是处理输入/输出的软件  
 B. 所有设备的启动工作都由系统统一来做  
 C. 来自通道的 I/O 中断事件由设备管理负责处理  
 D. 编制好的通道程序是存放在主存中的
17. 【2010 统考真题】本地用户通过键盘登录系统时，首先获得键盘输入信息的程序是（ ）。
- A. 命令解释程序      B. 中断处理程序  
 C. 系统调用服务程序      D. 用户登录程序
18. I/O 中断是 CPU 与通道协调工作的一种手段，所以在（ ）时，便要产生中断。
- A. CPU 执行“启动 I/O”指令而被通道拒绝接收  
 B. 通道接收了 CPU 的启动请求  
 C. 通道完成了通道程序的执行  
 D. 通道在执行通道程序的过程中
19. 一个计算机系统配置了 2 台绘图机和 3 台打印机，为了正确驱动这些设备，系统应该提供（ ）个设备驱动程序。
- A. 5      B. 3      C. 2      D. 1
20. 将系统调用参数翻译成设备操作命令的工作由（ ）完成。
- A. 用户层 I/O      B. 设备无关的操作系统软件  
 C. 中断处理      D. 设备驱动程序
21. 【2017 统考真题】系统将数据从磁盘读到内存的过程包括以下操作：
- ① DMA 控制器发出中断请求  
 ② 初始化 DMA 控制器并启动磁盘  
 ③ 从磁盘传输一块数据到内存缓冲区  
 ④ 执行“DMA 结束”中断服务程序
- 正确的执行顺序是（ ）。
- A. ③→①→②→④      B. ②→③→①→④  
 C. ②→①→③→④      D. ①→②→④→③
22. 【2011 统考真题】用户程序发出磁盘 I/O 请求后，系统的正确处理流程是（ ）。
- A. 用户程序→系统调用处理程序→中断处理程序→设备驱动程序  
 B. 用户程序→系统调用处理程序→设备驱动程序→中断处理程序  
 C. 用户程序→设备驱动程序→系统调用处理程序→中断处理程序  
 D. 用户程序→设备驱动程序→中断处理程序→系统调用处理程序
23. 【2012 统考真题】操作系统的 I/O 子系统通常由 4 个层次组成，每层明确定义了与邻近层次的接口，其合理的层次组织排列顺序是（ ）。
- A. 用户级 I/O 软件、设备无关软件、设备驱动程序、中断处理程序

- B. 用户级 I/O 软件、设备无关软件、中断处理程序、设备驱动程序  
 C. 用户级 I/O 软件、设备驱动程序、设备无关软件、中断处理程序  
 D. 用户级 I/O 软件、中断处理程序、设备无关软件、设备驱动程序
24. 【2013 统考真题】用户程序发出磁盘 I/O 请求后，系统的处理流程是：用户程序→系统调用处理程序→设备驱动程序→中断处理程序。其中，计算数据所在磁盘的柱面号、磁头号、扇区号的程序是（ ）。
- A. 用户程序                              B. 系统调用处理程序  
 C. 设备驱动程序                            D. 中断处理程序
25. 一个典型的文本打印页面有 50 行，每行 80 个字符，假定一台标准的打印机每分钟能打印 6 页，向打印机的输出寄存器中写一个字符的时间很短，可忽略不计。若每打印一个字符都需要花费  $50\mu s$  的中断处理时间（包括所有服务），使用中断驱动 I/O 方式运行这台打印机，中断的系统开销占 CPU 的百分比为（ ）。
- A. 2%                                      B. 5%                                    C. 20%                                    D. 50%

## 二、综合应用题

1. DMA 方式与中断控制方式的主要区别是什么？
2. DMA 方式与通道方式的主要区别是什么？
3. 在一个 32 位 100MHz 的单总线计算机系统中（每 10ns 一个周期），磁盘控制器使用 DMA 以 40MB/s 的速率从存储器中读出数据或向存储器写入数据。假设计算机在没有被周期挪用的情况下，在每个循环周期中读取并执行一个 32 位的指令。这样做，磁盘控制器使指令的执行速度降低了多少？
4. 某计算机系统中，时钟中断处理程序每次执行时间为 2ms（包括进程切换开销），若时钟中断频率为 60Hz，问 CPU 用于时钟中断处理的时间比率为多少？
5. 考虑 56kb/s 调制解调器的性能，驱动程序输出一个字符后就阻塞，当一个字符打印完毕后，产生一个中断通知阻塞的驱动程序，输出下一个字符，然后阻塞。若发消息、输出一个字符和阻塞的时间总和为 0.1ms，则由于处理调制解调器而占用的 CPU 时间比率是多少？假设每个字符有一个开始位和一个结束位，共占 10 位。

### 5.1.6 答案与解析

#### 一、单项选择题

1. B

可寻址是块设备的基本特征，A 选项不正确；共享设备是指一段时间内允许多个进程同时访问的设备，因此 C 选项不正确。分配共享设备是不会引起进程死锁的，D 选项不正确。

2. C

虚拟设备并不允许用户使用更多的物理设备，也与用户使用物理设备的标准化方式有关。允许用户程序不必全部装入主存便可使用系统中的设备，这同样不是虚拟设备考虑的内容，因此选择 C 选项。

3. D

DMA 方式主要用于块设备，磁盘是典型的块设备。这道题也要求读者了解什么是 I/O 控制方式，A、B、C 显然都不是 I/O 控制方式。

4. A

中断寄存器位于计算机主机；不存在 I/O 地址寄存器；编程空间一般是由体系结构和操作系统共同决定的。控制寄存器和状态寄存器分别用于接收上层发来的命令并存放设备状态信号，是设备控制器与上层的接口；至于控制命令，每种设备对应的设备控制器都对应一组相应的控制命令，CPU 通过控制命令控制设备控制器。

5. C

接口用来传输信号，I/O 逻辑即设备控制器，用来实现对设备的控制。

6. D

设备映射表中记录了逻辑设备所对应的物理设备，体现了两者的对应关系。对设备映射表来说，不能实现具体的功能及管理物理设备。

7. A

DMA 是一种不经过 CPU 而直接从主存存取数据的数据交换模式，它在 I/O 设备和主存之间建立了一条直接数据通路，如磁盘。当然，这条数据通路只是逻辑上的，实际并未直接建立一条物理线路，而通常是通过总线进行的。

8. A

设置通道后，CPU 只需向通道发送一条 I/O 指令。通道在收到该指令后，便从内存中取出本次要执行的通道程序，然后执行该通道程序，仅当通道完成规定的 I/O 任务后，才向 CPU 发出中断信号。因此通道用于完成内存与外设的信息交换。

9. A

通道是一种特殊的处理器，所以属于硬件技术。SPOOLing、缓冲池、内存覆盖都是在内存的基础上通过软件实现的。

10. C

在 DMA 方式中，设备和内存之间可以成批地进行数据交换而不用 CPU 干预，CPU 只参与预处理和结束过程。

11. D

命令/状态寄存器控制 DMA 的工作模式并给 CPU 反映它当前的状态，地址寄存器存放 DMA 作业时的源地址和目标地址，数据寄存器存放要 DMA 转移的数据，只有堆栈指针寄存器不需要在 DMA 控制器中存放。

12. B

字节多路通道，它通常含有许多非分配型子通道，其数量可达几十到几百个，每个通道连接一台 I/O 设备，并控制该设备的 I/O 操作。这些子通道按时间片轮转方式共享主通道。各个通道循环使用主通道，各个通道每次完成其 I/O 设备的一个字节的交换，然后让出主通道的使用权。这样，只要字节多路通道扫描每个子通道的速率足够快，而连接到子通道上的设备的速率不太高时，便不至于丢失信息。

13. A

设备的固有属性决定了设备的使用方式；设备独立性可以提高设备分配的灵活性和设备的利用率；设备安全性可以保证分配设备时不会导致永久阻塞。设备分配时一般不需要考虑及时性。

14. A

计算机系统为每台设备确定一个编号以便区分和识别设备，这个确定的编号称为设备的绝对号。

15. C

三者的控制关系是层层递进的，只有 C 选项正确。

16. A

通道为特殊的处理器，所以不属于软件。其他几项均正确。

17. B

键盘是典型的通过中断 I/O 方式工作的外设，当用户输入信息时，计算机响应中断并通过中断处理程序获得输入信息。

18. C

CPU 启动通道时不管启动成功与否，通道都要回答 CPU，通道在执行通道程序的过程中，CPU 与通道并行，通道完成通道程序的执行后，便发 I/O 中断向 CPU 报告。

19. C

因为绘图机和打印机属于两种不同类型的设备，系统只要按设备类型配置设备驱动程序即可，即每类设备只需一个设备驱动程序。

20. B

系统调用命令是操作系统提供给用户程序的通用接口，不会因为具体设备的不同而改变。而设备驱动程序负责执行操作系统发出的 I/O 命令，它因设备不同而不同。

21. B

在开始 DMA 传输时，主机向内存写入 DMA 命令块，向 DMA 控制器写入该命令块的地址，启动 I/O 设备。然后，CPU 继续其他工作，DMA 控制器则继续直接操作内存总线，将地址放到总线上开始传输。整个传输完成后，DMA 控制器中断 CPU。因此执行顺序是 2, 3, 1, 4，选 B。

22. B

输入/输出软件一般从上到下分为 4 个层次：用户层、与设备无关的软件层、设备驱动程序及中断处理程序。与设备无关的软件层也就是系统调用的处理程序。

当用户使用设备时，首先在用户程序中发起一次系统调用，操作系统的内核接到该调用请求后，请求调用处理程序进行处理，再转到相应的设备驱动程序，当设备准备好或所需数据到达后，设备硬件发出中断，将数据按上述调用顺序逆向回传到用户程序中。

23. A

考查内容同上题。设备管理软件一般分为 4 个层次：用户层、与设备无关的系统调用处理层、设备驱动程序及中断处理程序。

24. C

计算磁盘号、磁头号和扇区号的工作是由设备驱动程序完成的。题中的功能因设备硬件的不同而不同，因此应由厂家提供的设备驱动程序实现。

25. A

这台打印机每分钟打印  $50 \times 80 \times 6 = 24000$  个字符，即每秒打印 400 个字符。每个字符打印中断需要占用 CPU 时间  $50\mu s$ ，所以每秒用于中断的系统开销为  $400 \times 50\mu s = 20ms$ 。若使用中断驱动 I/O，则 CPU 剩余的 980ms 可用于其他处理，中断的开销占 CPU 的 2%。因此，使用中断驱动 I/O 方式运行这台打印机是有意义的。

## 二、综合应用题

### 1. 解答：

DMA 控制方式与中断控制方式的主要区别如下：

- 1) 中断控制方式在每个数据传送完成后中断 CPU，而 DMA 控制方式则在所要求传送的一批数据全部传送结束时中断 CPU。
- 2) 中断控制方式的数据传送在中断处理时由 CPU 控制完成，而 DMA 控制方式则在 DMA

控制器的控制下完成。不过，在 DMA 控制方式中，数据传送的方向、存放数据的内存始址及传送数据的长度等仍然由 CPU 控制。

3) DMA 方式以存储器为核心，中断控制方式以 CPU 为核心。因此 DMA 方式能与 CPU 并行工作。

4) DMA 方式传输批量的数据，中断控制方式的传输则以字节为单位。

## 2. 解答：

在 DMA 控制方式中，在 DMA 控制器控制下设备和主存之间可以成批地进行数据交换而不用 CPU 干预，这样既减轻了 CPU 的负担，又大大提高了 I/O 数据传送的速度。通道控制方式与 DMA 控制方式类似，也是一种以内存为中心实现设备与内存直接交换数据的控制方式。不过在通道控制方式中，CPU 只需发出启动指令，指出通道相应的操作和 I/O 设备，该指令就可以启动通道并使通道从内存中调出相应的通道程序执行。与 DMA 控制方式相比，通道控制方式所需的 CPU 干预更少，并且一个通道可以控制多台设备，进一步减轻了 CPU 的负担。另外，对通道来说，可以使用一些指令灵活改变通道程序，这一点 DMA 控制方式无法做到。

## 3. 解答：

在 32 位单总线的系统中，磁盘控制器使用 DMA 传输数据的速率为 40MB/s，即每 100ns 传输 4B(32 位)的数据。控制器每读取 10 个指令就挪用一个周期。因此，磁盘控制器使指令的执行速度降低了 10%。

## 4. 解答：

时钟中断频率为 60Hz，因此中断周期为 1/60s，每个时钟周期中用于中断处理的时间为 2ms，因此比率为  $0.002/(1/60) = 12\%$ 。

## 5. 解答：

因为一个字符占 10 位，因此在 56kb/s 的速率下，每秒传送  $56000/10 = 5600$  个字符，即产生 5600 次中断。每次中断需 0.1ms，因此处理调制解调器占用的 CPU 时间共为  $5600 \times 0.1\text{ms} = 560\text{ms}$ ，占 56% 的 CPU 时间。

## 5.2 I/O 核心子系统

在学习本节时，请读者思考以下问题：

- 1) 当处理机和外部设备速度差距较大时，并且此时不想让其中一方等待，有什么办法可以解决问题？
- 2) 什么是设备的独立性？引入设备的独立性有什么好处？

### 5.2.1 I/O 子系统概述

由于 I/O 设备种类繁多，功能和传输速率差异巨大，因此需要多种方法来进行设备控制。这些方法共同组成了操作系统内核的 I/O 子系统，它将内核的其他方面从繁重的 I/O 设备管理中解放出来。I/O 核心子系统提供的服务主要有 I/O 调度、缓冲与高速缓存、设备分配与回收、假脱机、设备保护和差错处理等。

### 5.2.2 I/O 调度概念

I/O 调度就是确定一个好的顺序来执行这些 I/O 请求。应用程序所发布的系统调用的顺序不

一定总是最佳选择，所以需要 I/O 调度来改善系统整体性能，使进程之间公平地共享设备访问，减少 I/O 完成所需要的平均等待时间。

操作系统开发人员通过为每个设备维护一个请求队列来实现调度。当一个应用程序执行阻塞 I/O 系统调用时，该请求就加到相应设备的队列上。I/O 调度会重新安排队列顺序，以改善系统总体效率和应用程序的平均响应时间。

I/O 子系统还可使用主存或磁盘上的存储空间的技术，如缓冲、高速缓存、假脱机等来改善计算机效率。

4.3 节的磁盘调度算法其实就是 I/O 调度的一种。

### 5.2.3 高速缓存与缓冲区

#### 1. 磁盘高速缓存 (Disk Cache)

操作系统中使用磁盘高速缓存技术来提高磁盘的 I/O 速度，对高速缓存复制的访问要比原始数据访问更为高效。例如，正在运行的进程的指令既存储在磁盘上，又存储在物理内存上，也被复制到 CPU 的二级和一级高速缓存中。

不过，磁盘高速缓存技术不同于通常意义上的介于 CPU 与内存之间的小容量高速存储器，而是指利用内存中的存储空间来暂存从磁盘中读出的一系列盘块中的信息。因此，磁盘高速缓存逻辑上属于磁盘，物理上则是驻留在内存中的盘块。

高速缓存在内存中分为两种形式：一种是在内存中开辟一个单独的存储空间作为磁盘高速缓存，大小固定；另一种是把未利用的内存空间作为一个缓冲池，供请求分页系统和磁盘 I/O 时共享。

#### 2. 缓冲区 (Buffer)

在设备管理子系统中，引入缓冲区的主要目的如下：

- 1) 缓和 CPU 与 I/O 设备间速度不匹配的矛盾。
- 2) 减少对 CPU 的中断频率，放宽对 CPU 中断响应时间的限制。
- 3) 解决基本数据单元大小（即数据粒度）不匹配的问题。
- 4) 提高 CPU 和 I/O 设备之间的并行性。

其实现方法如下：

- 1) 采用硬件缓冲器，但由于成本太高，除一些关键部位外，一般不采用硬件缓冲器。
- 2) 采用缓冲区（位于内存区域）。

缓冲区有一个特点，即当缓冲区的数据非空时，不能往缓冲区冲入数据，只能从缓冲区把数据传出；当缓冲区为空时，可以往缓冲区冲入数据，但必须把缓冲区充满后，才能从缓冲区把数据传出。

根据系统设置缓冲器的个数，缓冲技术可以分为如下几种：

- 1) 单缓冲。在设备和处理机之间设置一个缓冲区。设备和处理机交换数据时，先把被交换数据写入缓冲区，然后需要数据的设备或处理机从缓冲区取走数据。

如图 5.5 所示，在块设备输入时，假定从磁盘把一块数据输入缓冲区的时间为  $T$ ，操作系统将该缓冲区中的数据传送到用户区的时间为  $M$ ，而 CPU 对这一块数据处理的时间为  $C$ 。在研究各种缓冲技术的每块数据的处理时间时，有一个技巧：假设一种初始状态，然后计算下一次到达相同状态时所需要的时间，就是处理一块数据所需要的时间。在单缓冲中，这种初始状态为：工作区是满的，缓冲区是空的。如题目没有明确说明，一般认为缓冲区的大小和工作区的大小相等。

我们假设  $T > C$ ，从初始状态开始，当工作区数据处理完后，时间为  $C$ ，缓冲区还没充满，

当缓冲区充满时, 经历了  $T$  时间, 停止再冲入数据, 然后缓冲区向工作区传送数据, 当工作区满了以后, 缓冲区的数据同时也为空, 用时为  $M$ , 到达下一个开始状态, 整个过程用时  $M + T$ ; 若  $T < C$ , 同理, 整个过程用时  $M + C$ 。所以单缓冲区处理每块数据的用时为  $\max(C, T) + M$ 。

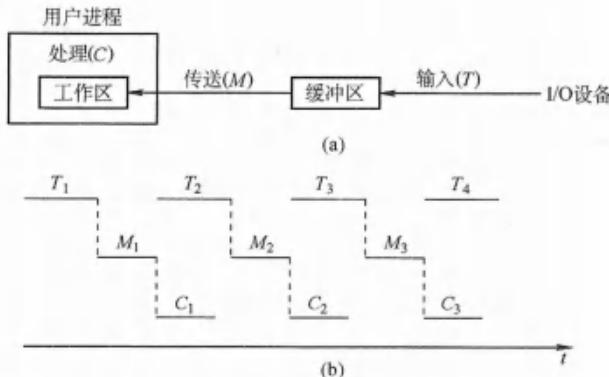


图 5.5 单缓冲工作示意图

2) 双缓冲。根据单缓冲的特点, CPU 在传送时间  $M$  内处于空闲状态, 由此引入双缓冲。I/O 设备输入数据时先装填到缓冲区 1, 在缓冲区 1 填满后才开始装填缓冲区 2, 与此同时处理器可以从缓冲区 1 中取出数据放入用户进程处理, 当缓冲区 1 中的数据处理完后, 若缓冲区 2 已填满, 则处理器又从缓冲区 2 中取出数据放入用户进程处理, 而 I/O 设备又可以装填缓冲区 1。注意, 必须等缓冲区 2 充满才能让处理器从缓冲区 2 取出数据。双缓冲机制提高了处理器和输入设备的并行操作的程度。

为了研究双缓冲处理一块数据的用时, 我们先规定一种初始状态: 工作区是空的, 其中一个缓冲区是满的, 另外一个缓冲区是空的; 我们不妨假设缓冲区 1 是空的, 缓冲区 2 是满的。

如图 5.6 所示, 我们假设  $T < C + M$ , 缓冲区 2 开始向工作区传送数据, 缓冲区 1 开始冲入数据, 当工作区充满数据后, 缓冲区为空, 时间为  $M$ , 然后工作区开始处理数据, 缓冲区 1 继续冲入数据, 因为此时只有一个 I/O 设备, 所以缓冲区 2 虽然为空, 但不能冲入数据。当缓冲区 1 充满数据后, 工作区的数据还未处理完毕, 时间为  $T$ , 当工作区数据处理完毕后, 此时工作区为空, 缓冲区 1 满, 缓冲区 2 空, 达到下一个初始状态, 用时  $C + M$ 。

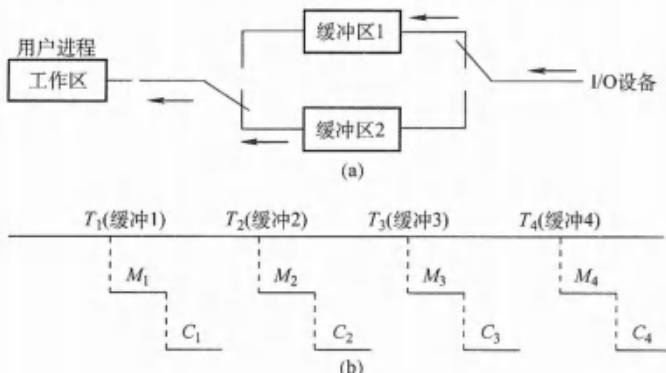


图 5.6 双缓冲工作示意图

我们再来分析  $T > C + M$  的情况。缓冲区 2 开始向工作区传送数据，缓冲区 1 开始冲入数据，当工作区充满数据并处理完后，用时  $C + M$ ，但缓冲区 1 的数据还未充满；当时间为  $T$  时，缓冲区 1 的数据充满，到达下一个初始状态。

总结：双缓冲区处理一块数据的用时为  $\max(C + M, T)$ 。

若  $M + C < T$ ，则可使块设备连续输入；若  $C + M > T$ ，则可使 CPU 不必等待设备输入。对于字符设备，若采用行输入方式，则采用双缓冲可使用户在输入第一行后，在 CPU 执行第一行中的命令的同时，用户可继续向第二缓冲区输入下一行数据。而单缓冲情况下则必须等待一行数据被提取完毕才可输入下一行的数据。

若两台机器之间通信仅配置了单缓冲，如图 5.7(a)所示，则它们在任意时刻都只能实现单方向的数据传输。例如，只允许把数据从 A 机传送到 B 机，或从 B 机传送到 A 机，而绝不允许双方同时向对方发送数据。为了实现双向数据传输，必须在两台机器中都设置两个缓冲区，一个用作发送缓冲区，另一个用作接收缓冲区，如图 5.7(b)所示。

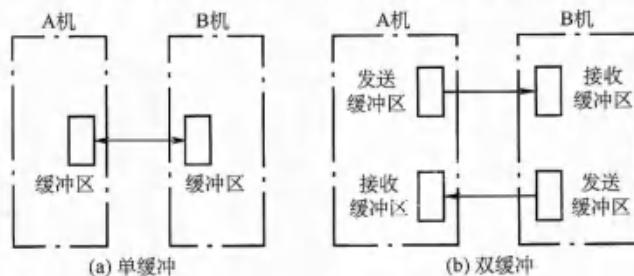


图 5.7 双机通信时缓冲区的设置

- 3) 循环缓冲。包含多个大小相等的缓冲区，每个缓冲区中有一个链接指针指向下一个缓冲区，最后一个缓冲区指针指向第一个缓冲区，多个缓冲区构成一个环形。

循环缓冲用于输入/输出时，还需要有两个指针  $in$  和  $out$ 。对输入而言，首先要从设备接收数据到缓冲区中， $in$  指针指向可以输入数据的第一个空缓冲区；当运行进程需要数据时，从循环缓冲区中取一个装满数据的缓冲区，并从此缓冲区中提取数据， $out$  指针指向可以提取数据的第一个满缓冲区。输出则正好相反。

- 4) 缓冲池。由多个系统公用的缓冲区组成，缓冲区按其使用状况可以形成三个队列：空缓冲队列、装满输入数据的缓冲队列（输入队列）和装满输出数据的缓冲队列（输出队列）。还应具有 4 种缓冲区：用于收容输入数据的工作缓冲区、用于提取输入数据的工作缓冲区、用于收容输出数据的工作缓冲区及用于提取输出数据的工作缓冲区，如图 5.8 所示。

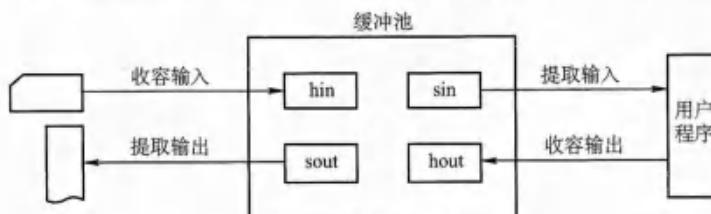


图 5.8 缓冲池的工作方式

当输入进程需要输入数据时，便从空缓冲队列的队首摘下一个空缓冲区，把它作为收容输入工作缓冲区，然后把输入数据输入其中，装满后再将它挂到输入队列队尾。当计算进程需要输入数据时，便从输入队列取得一个缓冲区作为提取输入工作缓冲区，计算进程从中提取数据，数据

用完后再将它挂到空缓冲队列尾。当计算进程需要输出数据时，便从空缓冲队列的队首取得一个空缓冲区，作为收容输出工作缓冲区，当其中装满输出数据后，再将它挂到输出队列队尾。当要输出时，由输出进程从输出队列中取得一个装满输出数据的缓冲区，作为提取输出工作缓冲区，当数据提取完后，再将它挂到空缓冲队列的队尾。

对于循环缓冲和缓冲池，我们只是定性地介绍它们的机理，而不去定量研究它们平均处理一块数据所需要的时间。而对于单缓冲和双缓冲，我们只要按照上面的模板分析，就可以解决任何计算单缓冲和双缓冲情况下数据块处理时间的问题，以不变应万变。

### 3. 高速缓存与缓冲区的对比

高速缓存是可以保存数据拷贝的高速存储器，访问高速缓存比访问原始数据更高效，速度更快。高速缓存和缓冲区的对比见表 5.1。

表 5.1 高速缓存和缓冲区的对比

		高速缓存	缓冲区
相同点		都介于高速设备和低速设备之间	
区别	存放数据	存放的是低速设备上的某些数据的复制数据，即高速缓存上有的，低速设备上面必然有	存放的是低速设备传递给高速设备的数据（或相反），而这些数据在低速设备（或高速设备）上却不一定有备份，这些数据再从缓冲区传送到高速设备（或低速设备）
	目的	高速缓存存放的是高速设备经常要访问的数据，若高速设备要访问的数据不在高速缓存中，则高速设备就需要访问低速设备	高速设备和低速设备的通信都要经过缓冲区，高速设备永远不会直接去访问低速设备

## 5.2.4 设备分配与回收

### 1. 设备分配概述

设备分配是指根据用户的 I/O 请求分配所需的设备。分配的总原则是充分发挥设备的使用效率，尽可能地让设备忙碌，又要避免由于不合理的分配方法造成进程死锁。从设备的特性来看，采用下述三种使用方式的设备分别称为独占设备、共享设备和虚拟设备。

- 1) 独占式使用设备。指在申请设备时，若设备空闲，则将其独占，不再允许其他进程申请使用，一直等到该设备被释放才允许其他进程申请使用。例如，打印机，在使用它打印时，只能独占式使用，否则在同一张纸上交替打印不同任务的内容，无法正常阅读。
- 2) 分时式共享使用设备。独占式使用设备时，设备利用率很低，当设备没有独占使用的请求时，可以通过分时共享使用提高利用率。例如，对磁盘设备的 I/O 操作，各进程的每次 I/O 操作请求可以通过分时来交替进行。
- 3) 以 SPOOLing 方式使用外部设备。SPOOLing (Simultaneous Peripheral Operation On-Line) 技术是在批处理操作系统时代引入的，即假脱机 I/O 技术。这种技术用于对设备的操作，实质上就是对 I/O 操作进行批处理。SPOOLing 技术实质上是一种以空间换时间的技术，而我们熟悉的请求分页系统中的页面调度算法就刚好相反，是以时间换空间的技术。

### 2. 设备分配的数据结构

设备分配依据的主要数据结构有设备控制表 (DCT)、控制器控制表 (COCT)、通道控制表 (CHCT) 和系统设备表 (SDT)，各数据结构功能如下。

设备控制表 (DCT)：我们可以认为，一个设备控制表就表征一个设备，而这个控制表中的表项就是设备的各个属性，如图 5.9 所示。

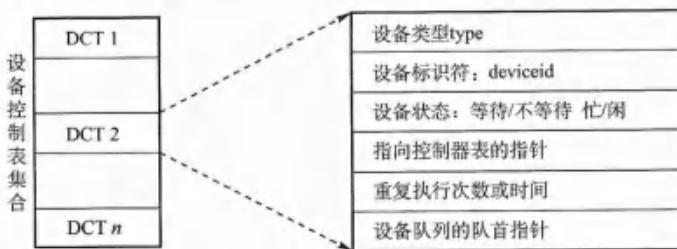


图 5.9 设备控制表

前面我们学过 4 种 I/O 控制方式，通道方式显然要比其他几种方式更加优越，因此现代操作系统的 I/O 控制采用的都是通道控制。设备控制器控制设备与内存交换数据，而设备控制器又需要请求通道为它服务，因此每个 COCT [见图 5.10(a)] 必定有一个表项存放指向相应通道控制表 (CHCT) [见图 5.10(b)] 的指针，而一个通道可为多个设备控制器服务，因此 CHCT 中必定有一个指针，指向一个表，这个表上的信息表达的是 CHCT 提供服务的那几个设备控制器。CHCT 与 COCT 的关系是一对多的关系。

系统设备表 (SDT): 整个系统只有一张 SDT，如图 5.10(c) 所示。它记录已连接到系统中的所有物理设备的情况，每个物理设备占一个表目。

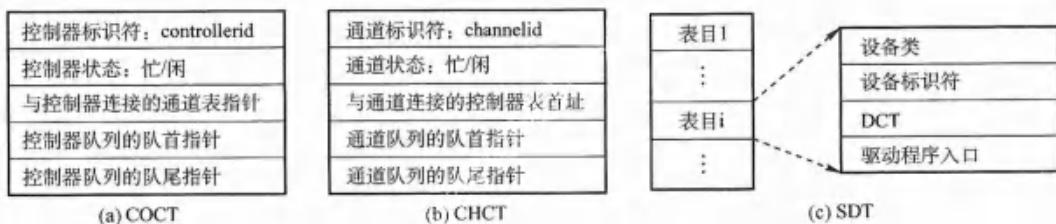


图 5.10 COCT、CHCT 和 SDT

由于在多道程序系统中，进程数多于资源数，会引起资源的竞争，因此要有一套合理的分配原则，主要考虑的因素有：I/O 设备的固有属性、I/O 设备的分配算法、I/O 设备分配的安全性以及 I/O 设备的独立性。

### 3. 设备分配的策略

- 1) 设备分配原则。设备分配应根据设备特性、用户要求和系统配置情况。分配的总原则是：既要充分发挥设备的使用效率，又要避免造成进程死锁，还要将用户程序和具体设备隔离开。
- 2) 设备分配方式。设备分配方式有静态分配和动态分配两种。

静态分配主要用于对独占设备的分配，它在用户作业开始执行前，由系统一次性分配该作业所要求的全部设备、控制器（如通道等）。一旦分配，这些设备、控制器（和通道）就一直为该作业所占用，直到该作业被撤销。静态分配方式不会出现死锁，但设备的使用效率低。因此，静态分配方式并不符合分配的总原则。

动态分配在进程执行过程中根据执行需要进行。当进程需要设备时，通过系统调用命令向系统提出设备请求，由系统按照事先规定的策略给进程分配所需要的设备、I/O 控制器，一旦用完，便立即释放。动态分配方式有利于提高设备的利用率，但若分配算法使用不当，则有可能造成进程死锁。

3) 设备分配算法。常用的动态设备分配算法有先请求先分配、优先级高者优先等。

对于独占设备，既可以采用动态分配方式，又可以采用静态分配方式，但往往采用静态分配方式，即在作业执行前，将作业所要用的这一类设备分配给它。共享设备可被多个进程所共享，一般采用动态分配方式，但在每个I/O传输的单位时间内只被一个进程所占有，通常采用先请求先分配和优先级高者优先的分配算法。

#### 4. 设备分配的安全性

设备分配的安全性是指设备分配中应防止发生进程死锁。

1) 安全分配方式。每当进程发出I/O请求后便进入阻塞态，直到其I/O操作完成时才被唤醒。

这样，一旦进程已经获得某种设备后便阻塞，不能再请求任何资源，而且在它阻塞时也不保持任何资源。优点是设备分配安全；缺点是CPU和I/O设备是串行工作的（对同一进程而言）。

2) 不安全分配方式。进程在发出I/O请求后继续运行，需要时又发出第二个、第三个I/O请求等。仅当进程所请求的设备已被另一进程占用时，才进入阻塞态。优点是一个进程可同时操作多个设备，从而迅速推进进程；缺点是这种设备分配有可能产生死锁。

#### 5. 逻辑设备名到物理设备名的映射

为了提高设备分配的灵活性和设备的利用率，方便实现I/O重定向，引入了设备独立性。设备独立性是指应用程序独立于具体使用的物理设备。

为了实现设备独立性，在应用程序中使用逻辑设备名来请求使用某类设备，在系统中设置一张逻辑设备表（Logical Unit Table, LUT），用于将逻辑设备名映射为物理设备名。LUT表项包括逻辑设备名、物理设备名和设备驱动程序入口地址；当进程用逻辑设备名来请求分配设备时，系统为它分配相应的物理设备，并在LUT中建立一个表项，以后进程再利用逻辑设备名请求I/O操作时，系统通过查找LUT来寻找相应的物理设备和驱动程序。

在系统中可采取两种方式建立逻辑设备表：

- 1) 在整个系统中只设置一张LUT。这样，所有进程的设备分配情况都记录在这张表中，因此不允许有相同的逻辑设备名，主要适用于单用户系统。
- 2) 为每个用户设置一张LUT。当用户登录时，系统便为该用户建立一个进程，同时也为之建立一张LUT，并把该表放入进程的PCB。

#### 5.2.5 SPOOLing技术（假脱机技术）

为了缓和CPU的高速性与I/O设备低速性之间的矛盾，引入了脱机输入/输出技术。该技术利用专门的外围控制机，将低速I/O设备上的数据传送到高速磁盘上，或者相反。SPOOLing的意思是外部设备同时联机操作，又称假脱机输入/输出操作，是操作系统中采用的一项将独占设备改造成共享设备的技术。

SPOOLing系统的组成如图5.11所示。

##### 1. 输入井和输出井

输入井和输出井是指在磁盘上开辟出的两个存储区域。输入井模拟脱机输入时的磁盘，



图5.11 SPOOLing系统的组成

用于收容 I/O 设备输入的数据。输出井模拟脱机输出时的磁盘，用于收容用户程序的输出数据。

## 2. 输入缓冲区和输出缓冲区

输入缓冲区和输出缓冲区是在内存中开辟的两个缓冲区。输入缓冲区用于暂存由输入设备送来的数据，以后再传送到输入井。输出缓冲区用于暂存从输出井送来的数据，以后再传送到输出设备。

## 3. 输入进程和输出进程

输入进程模拟脱机输入时的外围控制机，将用户要求的数据从输入机通过输入缓冲区再送到输入井。当 CPU 需要输入数据时，直接将数据从输入井读入内存。输出进程模拟脱机输出时的外围控制机，把用户要求输出的数据先从内存送到输出井，待输出设备空闲时，再将输出井中的数据经过输出缓冲区送到输出设备。

共享打印机是使用 SPOOLing 技术的一个实例，这项技术已被广泛地用于多用户系统和局域网络。当用户进程请求打印输出时，SPOOLing 系统同意为它打印输出，但并不真正立即把打印机分配给该用户进程，而只为它做两件事：

- 1) 由输出进程在输出井中为之申请一个空闲磁盘块区，并将要打印的数据送入其中。
- 2) 输出进程再为用户进程申请一张空白的用户请求打印表，并将用户的打印要求填入其中，再将该表挂到请求打印队列上。

SPOOLing 系统的主要特点有：提高了 I/O 的速度；将独占设备改造为共享设备；实现了虚拟设备功能。

前面我们提到过 SPOOLing 技术是一种以空间换时间的技术，我们很容易理解它牺牲了空间，因为它开辟了磁盘上的空间作为输入井和输出井，但它又如何节省时间呢？

从前述内容我们了解到，磁盘是一种高速设备，在与内存交换数据的速度上优于打印机、键盘、鼠标等中低速设备。试想一下，若没有 SPOOLing 技术，CPU 要向打印机输出要打印的数据，打印机的打印速度比较慢，CPU 就必须迁就打印机，在打印机把数据打印完后才能继续做其他的工作，浪费了 CPU 的不少时间。在 SPOOLing 技术下，CPU 要打印机打印的数据可以先输出到磁盘的输出井中（这个过程由输出进程控制），然后做其他的事情。若打印机此时被占用，则 SPOOLing 系统就会把这个打印请求挂到等待队列上，待打印机有空时再把数据打印出来。向磁盘输出数据的速度比向打印机输出数据的速度快，因此就节省了时间。

### 5.2.6 本节小结

本节开头提出的问题的参考答案如下。

1) 当处理机和外部设备速度差距较大时，并且此时不想让其中一方等待，有什么办法可以解决问题？

可以采用缓冲技术来缓解处理机与外部设备速度上的矛盾，即在某块地方（一般为主存）设立一片缓冲区，外部设备与处理机的输入/输出都经过缓冲区，这样外部设备和处理机就都不用互相等待。

2) 什么是设备的独立性？引入设备的独立性有什么好处？

设备独立性是指用户在编程序时使用的设备与实际设备无关。一个程序应独立于分配给它的某类设备的具体设备，即在用户程序中只指明 I/O 使用的设备类型即可。

设备独立性有以下优点：

- ① 方便用户编程。

- ② 使程序运行不受具体机器环境的限制。  
 ③ 便于程序移植。

### 5.2.7 本节习题精选

#### 一、单项选择题

1. 以下( )不属于设备管理数据结构。
 

A. PCB	B. DCT	C. COCT	D. CHCT
--------	--------	---------	---------
2. 设备的独立性是指( )。
 

A. 设备独立于计算机系统	B. 系统对设备的管理是独立的	C. 用户编程时使用的设备与实际使用的设备无关	D. 每台设备都有一个唯一的编号
---------------	-----------------	-------------------------	------------------
3. 下列( )不是设备的分配方式。
 

A. 独享分配	B. 共享分配	C. 虚拟分配	D. 分区分配
---------	---------	---------	---------
4. 下面设备中属于共享设备的是( )。
 

A. 打印机	B. 磁带机	C. 磁盘	D. 磁带机和磁盘
--------	--------	-------	-----------
5. 【2009统考真题】程序员利用系统调用打开I/O设备时，通常使用的设备标识是( )。
 

A. 逻辑设备名	B. 物理设备名	C. 主设备号	D. 从设备号
----------	----------	---------	---------
6. 引入高速缓冲的主要目的是( )。
 

A. 提高CPU的利用率	B. 提高I/O设备的利用率
C. 改善CPU与I/O设备速度不匹配的问题	D. 节省内存
7. 【2012统考真题】下列选项中，不能改善磁盘设备I/O性能的是( )。
 

A. 重排I/O请求次序	B. 在一个磁盘上设置多个分区
C. 预读和滞后写	D. 优化文件物理块的分布
8. 为了使并发进程能有效地进行输入和输出，最好采用( )结构的缓冲技术。
 

A. 缓冲池	B. 循环缓冲	C. 单缓冲	D. 双缓冲
--------	---------	--------	--------
9. 在采用SPOOLing技术的系统中，用户的打印结果首先被送到( )。
 

A. 磁盘固定区域	B. 内存固定区域	C. 终端	D. 打印机
-----------	-----------	-------	--------
10. 缓冲技术中的缓冲池在( )中。
 

A. 主存	B. 外存	C. ROM	D. 寄存器
-------	-------	--------	--------
11. 设从磁盘将一块数据传送到缓冲区所用的时间为80μs，将缓冲区中的数据传送到用户区所用的时间为40μs，CPU处理一块数据所用的时间为30μs。若有多块数据需要处理，并采用单缓冲区传送某磁盘数据，则处理一块数据所用的总时间为( )。
 

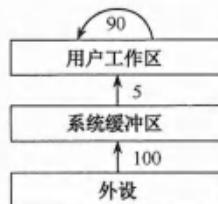
A. 120μs	B. 110μs	C. 150μs	D. 70μs
----------	----------	----------	---------
12. 某操作系统采用双缓冲区传送磁盘上的数据。设从磁盘将数据传送到缓冲区所用的时间为T<sub>1</sub>，将缓冲区中的数据传送到用户区所用的时间为T<sub>2</sub>(假设T<sub>2</sub>远小于T<sub>1</sub>)，CPU处理数据所用的时间为T<sub>3</sub>，则处理该数据，系统所用的总时间为( )。
 

A. T <sub>1</sub> +T <sub>2</sub> +T <sub>3</sub>	B. max(T <sub>2</sub> , T <sub>3</sub> )+T <sub>1</sub>	C. max(T <sub>1</sub> , T <sub>3</sub> )+T <sub>2</sub>	D. max(T <sub>1</sub> , T <sub>2</sub> +T <sub>3</sub> )
---	---	---	--
13. 若I/O所花费的时间比CPU的处理时间短得多，则缓冲区( )。
 

A. 最有效	B. 几乎无效
C. 均衡	D. 以上答案都不对

14. 【2011 统考真题】某文件占 10 个磁盘块，现要把该文件的磁盘块逐个读入主存缓冲区，并送用户区进行分析，假设一个缓冲区与一个磁盘块大小相同，把一个磁盘块读入缓冲区的时间为  $100\mu s$ ，将缓冲区的数据传送到用户区的时间是  $50\mu s$ ，CPU 对一块数据进行分析的时间为  $50\mu s$ 。在单缓冲区和双缓冲区结构下，读入并分析完该文件的时间分别是（ ）。

- A.  $1500\mu s, 1000\mu s$
  - B.  $1550\mu s, 1100\mu s$
  - C.  $1550\mu s, 1550\mu s$
  - D.  $2000\mu s, 2000\mu s$
15. 【2013 统考真题】设系统缓冲区和用户工作区均采用单缓冲，从外设读入一个数据块到系统缓冲区的时间为 100，从系统缓冲区读入一个数据块到用户工作区的时间为 5，对用户工作区中的一个数据块进行分析的时间为 90（如下图所示）。进程从外设读入并分析 2 个数据块的最短时间是（ ）。



- A. 200
  - B. 295
  - C. 300
  - D. 390
16. 缓冲区管理着要考虑的问题是（ ）。
- A. 选择缓冲区的大小
  - B. 决定缓冲区的数量
  - C. 实现进程访问缓冲区的同步
  - D. 限制进程的数量
17. 考虑单用户计算机上的下列 I/O 操作，需要使用缓冲技术的是（ ）。
- I. 图形用户界面下使用鼠标
  - II. 多任务操作系统下的磁带驱动器（假设没有设备预分配）
  - III. 包含用户文件的磁盘驱动器
  - IV. 使用存储器映射 I/O，直接和总线相连的图形卡
- A. I、III
  - B. II、IV
  - C. II、III、IV
  - D. 全选
18. 提高单机资源利用率的关键技术是（ ）。
- A. SPOOLing 技术
  - B. 虚拟技术
  - C. 交换技术
  - D. 多道程序设计技术
19. 虚拟设备是靠（ ）技术来实现的。
- A. 通道
  - B. 缓冲
  - C. SPOOLing
  - D. 控制器
20. SPOOLing 技术的主要目的是（ ）。
- A. 提高 CPU 和设备交换信息的速度
  - B. 提高独占设备的利用率
  - C. 减轻用户编程负担
  - D. 提供主、辅存接口
21. 采用 SPOOLing 技术的计算机系统，外围计算机需要（ ）。
- A. 一台
  - B. 多台
  - C. 至少一台
  - D. 0 台
22. SPOOLing 系统由（ ）组成。
- A. 预输入程序、井管理程序和缓输出程序
  - B. 预输入程序、井管理程序和井管理输出程序
  - C. 输入程序、井管理程序和输出程序
  - D. 预输入程序、井管理程序和输出程序

23. 在 SPOOLing 系统中，用户进程实际分配到的是( )。
- A. 用户所要求的外设
  - B. 外存区，即虚拟设备
  - C. 设备的一部分存储区
  - D. 设备的一部分空间
24. 下面关于 SPOOLing 系统的说法中，正确的是( )。
- A. 构成 SPOOLing 系统的基本条件是有外围输入机与外围输出机
  - B. 构成 SPOOLing 系统的基本条件是要有大容量、高速度的硬盘作为输入井和输出井
  - C. 当输入设备忙时，SPOOLing 系统中的用户程序暂停执行，待 I/O 空闲时再被唤醒执行输出操作
  - D. SPOOLing 系统中的用户程序可以随时将输出数据送到输出井中，待输出设备空闲时再由 SPOOLing 系统完成数据的输出操作
25. 下面关于 SPOOLing 的叙述中，不正确的是( )。
- A. SPOOLing 系统中不需要独占设备
  - B. SPOOLing 系统加快了作业执行的速度
  - C. SPOOLing 系统使独占设备变成共享设备
  - D. SPOOLing 系统提高了独占设备的利用率
26. ( ) 是操作系统中采用的以空间换取时间的技术。
- A. SPOOLing 技术
  - B. 虚拟存储技术
  - C. 覆盖与交换技术
  - D. 通道技术
27. 采用假脱机技术，将磁盘的一部分作为公共缓冲区以代替打印机，用户对打印机的操作实际上是对磁盘的存储操作，用以代替打印机的部分由( ) 完成。
- A. 独占设备
  - B. 共享设备
  - C. 虚拟设备
  - D. 一般物理设备
28. 下面关于独占设备和共享设备的说法中，不正确的是( )。
- A. 打印机、扫描仪等属于独占设备
  - B. 对独占设备往往采用静态分配方式
  - C. 共享设备是指一个作业尚未撤离，另一个作业即可使用，但每个时刻只有一个作业使用
  - D. 对共享设备往往采用静态分配方式
29. 在采用 SPOOLing 技术的系统中，用户的打印数据首先被送到( )。
- A. 磁盘固定区域
  - B. 内存固定区域
  - C. 终端
  - D. 打印机
30. 【2016 统考真题】下列关于 SPOOLing 技术的叙述中，错误的是( )。
- A. 需要外存的支持
  - B. 需要多道程序设计技术的支持
  - C. 可以让多个作业共享一台独占式设备
  - D. 由用户作业控制设备与输入/输出井之间的数据传送
31. 【2015 统考真题】在系统内存中设置磁盘缓冲区的主要目的是( )。
- A. 减少磁盘 I/O 次数
  - B. 减少平均寻道时间
  - C. 提高磁盘数据可靠性
  - D. 实现设备无关性
32. 【2020 统考真题】对于具备设备独立性的系统，下列叙述中，错误的是( )。
- A. 可以使用文件名访问物理设备
  - B. 用户程序使用逻辑设备名访问物理设备
  - C. 需要建立逻辑设备与物理设备之间的映射关系
  - D. 更换物理设备后必须修改访问该设备的应用程序

## 二、综合应用题

1. 用于设备分配的数据结构有哪些？它们之间的关系是什么？
2. 输入/输出软件一般分为 4 个层次：用户层、与设备无关的软件层、设备驱动程序和中断处理程序。请说明以下各工作是在哪一层完成的：
  - 1) 为磁盘读操作计算磁道、扇区和磁头。
  - 2) 向设备寄存器写命令。
  - 3) 检查用户是否有权使用设备。
  - 4) 将二进制证书转换成 ASCII 码以便打印。
3. 一个串行线能以最大  $50000\text{B/s}$  的速度接收输入。数据平均输入速率是  $20000\text{B/s}$ 。若用轮询来处理输入，不管是否有输入数据，轮询例程都需要  $3\mu\text{s}$  来执行。在下一个字节到达之前未从控制器中取走的字节将丢失。那么最大的安全的轮询时间间隔是多少？
4. 在某系统中，从磁盘将一块数据输入缓冲区需要花费的时间为  $T$ ，CPU 对一块数据进行处理的时间为  $C$ ，将缓冲区的数据传送到用户区所花的时间为  $M$ ，那么在单缓冲和双缓冲情况下，系统处理大量数据时，一块数据的处理时间为多少？
5. 在某系统中，若采用双缓冲区（每个缓冲区可存放一个数据块），将一个数据块从磁盘传送到缓冲区的时间为  $80\mu\text{s}$ ，从缓冲区传送到用户的时间为  $20\mu\text{s}$ ，CPU 计算一个数据块的时间为  $50\mu\text{s}$ 。总共处理 4 个数据块，每个数据块的平均处理时间是多少？
6. 一个 SPOOLing 系统由输入进程 I、用户进程 P、输出进程 O、输入缓冲区、输出缓冲区组成。进程 I 通过输入缓冲区为进程 P 输入数据，进程 P 的处理结果通过输出缓冲区交给进程 O 输出。进程间数据交换以等长度的数据块为单位。这些数据块均存储在同一磁盘上。因此，SPOOLing 系统的数据块通信原语保证始终满足  $i+o \leq \max$ ，其中  $\max$  为磁盘容量（以该数据块为单位）， $i$  为磁盘上输入数据块的总数， $o$  为磁盘上输出数据块的总数。该 SPOOLing 系统运行时：只要有输入数据，进程 I 终究会将它放入输入缓冲区；只要输入缓冲区有数据块，进程 P 终究会读入、处理，并产生结果数据，写到输出缓冲区；只要输出缓冲区有数据块，进程 O 终究会输出它。  
请说明该 SPOOLing 系统在什么情况下死锁。请说明如何修正约束条件以避免死锁，同时仍允许输入数据块和输出数据块均存储在同一个磁盘上。

### 5.2.8 答案与解析

#### 一、单项选择题

1. A

DCT 是设备控制表；COCT 是控制器控制表；CHCT 是通道控制表；PCB 是进程控制块，不属于设备管理的数据结构。

2. C

设备的独立性主要是指用户使用设备的透明性，即使用户程序和实际使用的物理设备无关。

3. D

设备的分配方式主要有独享分享、共享分配和虚拟分配，D 是内存的分配方式。

4. C

共享设备是指在一个时间间隔内可被多个进程同时访问的设备，只有磁盘满足。打印机在一个时间间隔内被多个进程访问时打印出来的文档就会乱；磁带机旋转到所需的读写位置需要较长时间，若一个时间间隔内被多个进程访问，磁带机就只能一直旋转，没时间读写。

5. A

用户程序对 I/O 设备的请求采用逻辑设备名，而程序实际执行时使用物理设备名，它们之间的转换是由设备无关软件层完成的。主设备和从设备是总线仲裁中的概念。

#### 6. C

CPU 与 I/O 设备执行速度通常是对等的，前者快、后者慢，通过高速缓冲技术来改善两者不匹配的问题。

#### 7. B

对于 A，重排 I/O 请求次序也就是进行 I/O 调度，使进程之间公平地共享磁盘访问，减少 I/O 完成所需要的平均等待时间。对于 C，缓冲区结合预读和滞后写技术对于具有重复性及阵发性的 I/O 进程改善磁盘 I/O 性能很有帮助。对于 D，优化文件物理块的分布可以减少寻找时间与延迟时间，从而提高磁盘性能。在一个磁盘上设置多个分区与改善设备 I/O 性能并无多大联系，相反还会带来处理的复杂性，降低利用率。

#### 8. A

缓冲池是系统的公用资源，可供多个进程共享，并且既能用于输入又能用于输出。其一般包含三种类型的缓冲：① 空闲缓冲区；② 装满输入数据的缓冲区；③ 装满输出数据的缓冲区。为了管理上的方便，可将相同类型的缓冲区链成一个队列。B、C、D 属专用缓冲。

#### 9. A

输入井和输出井是在磁盘上开辟的两大存储空间。输入井模拟脱机输入时的磁盘设备，用于暂存 I/O 设备输入的数据；输出井模拟脱机输出时的磁盘，用于暂存用户程序的输出数据。为了缓和 CPU，打印结果首先送到位于磁盘固定区域的输出井。

#### 10. A

输入井和输出井是在磁盘上开辟的存储空间，而输入/输出缓冲区则是在内存中开辟的，因为 CPU 速度比 I/O 设备高很多，缓冲池通常在主存中建立。

#### 11. A

采用单缓冲区传送数据时，设备与处理机对缓冲区的操作是串行的，当进行第  $i$  次读磁盘数据送至缓冲区时，系统再同时读出用户区中第  $i-1$  次数据进行计算，此两项操作可以并行，并与数据从缓冲区传送到用户区的操作串行进行，所以系统处理一块数据所用的总时间为  $\max(80\mu s, 30\mu s) + 40\mu s = 120\mu s$ 。

#### 12. D

若  $T_3 > T_1$ ，即 CPU 处理数据块比数据传送慢，意味着 I/O 设备可连续输入，磁盘将数据传送到缓冲区，再传送到用户区，与 CPU 处理数据可视为并行处理，时间的花费取决于 CPU 最大花费时间，则系统所用总时间为  $T_3$ 。若  $T_3 < T_1$ ，即 CPU 处理数据比数据传送快，此时 CPU 不必等待 I/O 设备，磁盘将数据传送到缓冲区，与缓冲区中数据传送到用户区及 CPU 数据处理可视为并行执行，则花费时间取决于磁盘将数据传送到缓冲区所用时间  $T_1$ 。所以选择 D 选项。

#### 13. B

缓冲区主要解决输入/输出速度比 CPU 处理的速度慢而造成数据积压的矛盾。所以当 I/O 花费的时间比 CPU 处理时间短很多时，缓冲区没有必要设置。

#### 14. B

在单缓冲区中，当上一个磁盘块从缓冲区读入用户区完成时，下一磁盘块才能开始读入，也就是当最后一块磁盘块读入用户区完毕时所用的时间为  $150 \times 10 = 1500\mu s$ ，加上处理最后一个磁盘块的时间  $50\mu s$ ，得  $1550\mu s$ 。双缓冲区中，不存在等待磁盘块从缓冲区读入用户区的问题，10 个磁盘块可以连续从外存读入主存缓冲区，加上将最后一个磁盘块从缓冲区送到用户区的传输时间  $50\mu s$  及处理时间  $50\mu s$ ，也就是  $100 \times 10 + 50 + 50 = 1100\mu s$ 。

15. C

数据块 1 从外设到用户工作区的总时间为 105，在这段时间中，数据块 2 未进行操作。在数据块 1 进行分析处理时，数据块 2 从外设到用户工作区的总时间为 105，这段时间是并行的。再加上数据块 2 进行处理的时间 90，总共是 300，答案为 C。

16. C

在缓冲机制中，无论是单缓冲、多缓冲还是缓冲池，由于缓冲区是一种临界资源，所以在使用缓冲区时都有一个申请和释放（即互斥）的问题需要考虑。

17. D

在鼠标移动时，若有高优先级的操作产生，为了记录鼠标活动的情况，必须使用缓冲技术，I 正确。由于磁盘驱动器和目标或源 I/O 设备间的吞吐量不同，必须采用缓冲技术，II 正确。为了能使数据从用户作业空间传送到磁盘或从磁盘传送到用户作业空间，必须采用缓冲技术，III 正确。为了便于多幅图形的存取及提高性能，缓冲技术是可以采用的，特别是在显示当前一幅图形又要得到下一幅图形时，应采用双缓冲技术，IV 正确。

综上所述，本题正确答案为 D。

18. D

在单机系统中，最关键的资源是处理器资源，最大化地提高处理器利用率，就是最大化地提高系统效率。多道程序设计技术是提高处理器利用率的关键技术，其他均为设备和内存的相关技术。

19. C

SPOOLing 技术是操作系统中采用的一种将独占设备改造为共享设备的技术。通过这种技术处理后的设备通常称为虚拟设备。

20. B

SPOOLing 技术可将独占设备改造为共享设备，其主要目的是提高系统资源/独占设备的利用率。

21. D

SPOOLing 技术需要使用磁盘空间（输入井和输出井）和内存空间（输入/输出缓冲区），不需要外围计算机的支持。

22. A

SPOOLing 系统主要包含三部分，即输入井和输出井、输入缓冲区和输出缓冲区以及输入进程和输出进程。这三部分由预输入程序、井管理程序和缓输出程序管理，以保证系统正常运行。

23. B

通过 SPOOLing 技术可将一台物理 I/O 设备虚拟为 I/O 设备，同样允许多个用户共享一台物理 I/O 设备，所以 SPOOLing 并不是将物理设备真的分配给用户进程。

24. D

构成 SPOOLing 系统的基本条件是不仅要有大容量、高速度的外存作为输入井和输出井，而且还要有 SPOOLing 软件，因此 A 错误、B 不够全面，同时利用 SPOOLing 技术提高了系统和 I/O 设备的利用率，进程不必等待 I/O 操作的完成，因此 C 选项也不正确。

25. A

因为 SPOOLing 技术是一种典型的虚拟设备技术，它通过将独占设备虚拟成共享设备，使得多个进程共享一个独占设备，从而加快作业的执行速度，提高独占设备的利用率。既然是将独占设备虚拟成共享设备，所以必须先有独占设备才行。

26. A

SPOOLing 技术需有高速大容量且可随机存取的外存支持，通过预输入及缓输出来减少 CPU 等待慢速设备的时间，将独享设备改造成共享设备。

27. C

打印机是独享设备，利用 SPOOLing 技术可将打印机改造为可供多个用户共享的虚拟设备。

28. D

独占设备采用静态分配方式，而共享设备采用动态分配方式。

29. A

用户的打印数据首先被送到输出井，输出井在磁盘中。

30. D

SPOOLing 利用专门的外围控制机，将低速 I/O 设备上的数据传送到高速磁盘上，或者相反。SPOOLing 的意思是外部设备同时联机操作，又称假脱机输入/输出操作，是操作系统中采用的一项将独占设备改造成共享设备的技术。高速磁盘即外存，A 正确。SPOOLing 技术需要进行输入/输出操作，单道批处理系统无法满足，B 正确。SPOOLing 技术实现了将独占设备改造成共享设备的技术，C 正确。设备与输入井/输出井之间数据的传送是由系统实现的，D 错误。

31. A

磁盘和内存的速度差异，决定了可以将内存经常访问的文件调入磁盘缓冲区，从高速缓存中复制的访问比磁盘 I/O 的机械操作要快很多。

32. D

设备可视为特殊文件，A 正确。用户使用逻辑设备名来访问物理文件，有利于设备独立性，B 正确。通过逻辑设备名访问物理设备时，需要建立逻辑设备和物理设备之间的映射关系，C 正确。应用程序按逻辑设备名访问设备，再经驱动程序的处理来控制物理设备，若更换物理设备，则只需更换驱动程序，而无须修改应用程序，D 错误。

## 二、综合应用题

1. 解答：

用于设备分配的数据结构有系统设备表(SDT)、设备控制表(DCT)、控制器控制表(COCT)和通道控制表(CHCT)。

SDT 整个系统中只有一张，它记录系统中全部设备的情况，是系统范围的数据结构。每个设备有一张 DCT，系统为每个设备配置一张 DCT，以记录本设备的情况。每个控制器有一张 COCT，系统为每个控制器都设置一张用于记录本控制器情况的 COCT。系统为每个通道配置一张 CHCT，以记录通道情况。SDT 中有一个 DCT 指针，DCT 中的每个表目都有一个指向 COCT 的指针，COCT 中有一个 CHCT 指针，CHCT 中有一个 COCT 指针。

2. 解答：

分析：首先，我们来看这些功能是不是应该由操作系统来完成。操作系统是一个代码相对稳定的软件，它很少发生代码的变化。若 1) 由操作系统完成，则操作系统就必须记录逻辑块和磁盘细节的映射，操作系统的代码会急剧膨胀，而且对新型介质的支持也会引起代码的变动。若 2) 也由操作系统完成，则操作系统需要记录不同生产厂商的不同数据，而且后续新厂商和新产品也无法得到支持。

因为 1) 和 2) 都与具体的磁盘类型有关，因此为了能够让操作系统尽可能多地支持各种不同型号的设备，1) 和 2) 应由厂商所编写的设备驱动程序完成。3) 涉及安全与权限问题，应由与设备无关的操作系统完成。4) 应由用户层来完成，因为只有用户知道将二进制整数转换为 ASCII 码的格式（使用二进制还是十进制、有没有特别的分隔符等）。

## 3. 解答：

串行线接收数据的最大速度为  $50000\text{B/s}$ , 即每  $20\mu\text{s}$  接收 1B, 而轮询例程需  $3\mu\text{s}$  来执行, 因此最大的安全轮询时间间隔是  $17\mu\text{s}$ 。

## 4. 解答：

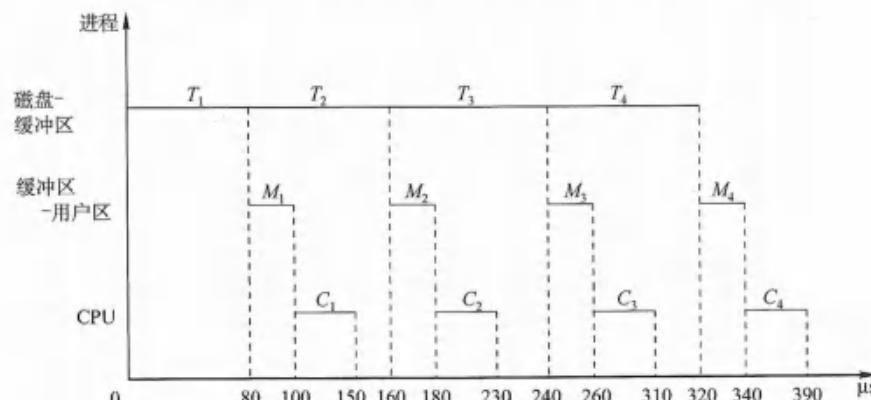
- 1) 在单缓冲的情况下, 应先从磁盘把一块数据输入缓冲区, 所花费的时间为  $T$ ; 然后由操作系统将缓冲区的数据传送到用户区, 所花的时间为  $M$ ; 接下来便由 CPU 对这一块数据进行计算, 计算时间为  $C$ 。由于 CPU 的计算操作与磁盘的数据输入操作可以并行, 因此一块数据的处理时间为  $\max(C, T) + M$ 。
- 2) 在双缓冲的情况下, 应先从磁盘把一块数据输入第一个缓冲区, 当装满第一个缓冲区后, 操作系统可将第一个缓冲区的数据传送到用户区并对第一块数据进行计算, 与此同时可将磁盘输入数据送入第二个缓冲区; 当计算完成后, 若第二个缓冲区已装满数据, 则又可以将第二个缓冲区中的数据传送至用户区并对第二块数据进行计算, 与此同时可将磁盘输入数据送入第一个缓冲区, 如此反复交替使用两个缓冲区。CPU 处理一个缓冲区中的数据的耗时为  $C + M$ , 而准备好另一个缓冲区内的数据的耗时为  $T$ 。因此, 当  $C + M > T$  时, CPU 刚处理完一个缓冲区的数据, 另一个缓冲区的数据就已经准备好了, 就可以紧接着处理下一块数据, 因此平均来看, 每处理一块数据耗时为  $C + M$ 。而当  $C + M < T$  时, CPU 处理完一个缓冲区的数据, 另一个缓冲区的数据还没准备好, 因此每隔  $T$  的时间, 才可以开始处理下一块数据, 平均来看处理一块数据耗时为  $T$ 。综上, 采用双缓冲区, 处理一块数据平均耗时为  $\max(C + M, T)$ 。

注意: 在无缓冲的情况下, 为了读取磁盘数据, 应先从磁盘把一块数据输入用户数据区, 所花费的时间为  $T$ ; 然后由 CPU 对一块数据进行计算, 计算时间为  $C$ , 所以每块数据的处理时间为  $T + C$ 。

## 5. 解答:

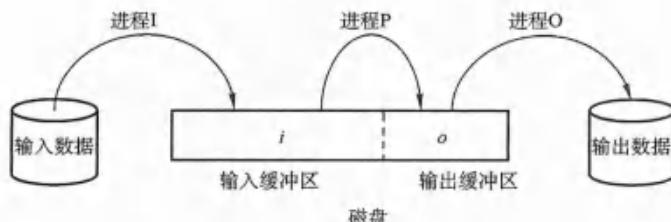
4 个数据块的处理过程如下图所示, 总耗时  $390\mu\text{s}$ , 每个数据块的平均处理时间为  $390\mu\text{s}/4 = 97.5\mu\text{s}$ 。

从中看到, 处理  $n$  个数据块的总耗时为  $(80n + 20 + 50)\mu\text{s} = (80n + 70)\mu\text{s}$ , 每个数据块的平均处理时间为  $(80n + 70)/n \mu\text{s}$ , 当  $n$  较大时, 平均时间近似为  $\max(C, T) = 80\mu\text{s}$ 。



## 6. 解答:

此系统的示意图如下图所示。



下面找到一种导致该SPOOLing系统死锁的情况：当磁盘上输入数据块总数  $i = \max$  时，磁盘上输出数据块的总数  $o$  必然为 0。此时，进程 I 发现输入缓冲区已满，所以不能再把输入数据放入缓冲区；进程 P 此时有一个处理完的数据，打算把结果数据放入缓冲区，但也发现没有空闲的空间可以放结果数据，因为  $o = 0$ ；所以没有输出数据可以输出，于是进程 O 也无事可做。这时进程 I、P、O 各自都等待着一个事件的发生，若没有外力的作用，它们将一直等待下去，这种僵局显然是死锁。只需将条件修改为  $i + o \leq \max$ ，且  $i \leq \max - 1$ ，就不会再发生死锁。

### 5.3 本章疑难点

- 1) 分配设备。首先根据 I/O 请求中的物理设备名查找系统设备表 (SDT)，从中找出该设备的 DCT，再根据 DCT 中的设备状态字段，可知该设备是否正忙。若忙，便将请求 I/O 进程的 PCB 挂到设备队列上；若空闲，则按照一定的算法计算设备分配的安全性，若安全则将设备分配给请求进程，否则仍将其 PCB 挂到设备队列上。
- 2) 分配控制器。系统把设备分配给请求 I/O 的进程后，再到其 DCT 中找出与该设备连接的控制器的 COCT，从 COCT 中的状态字段中可知该控制器是否忙碌。若忙，则将请求 I/O 进程的 PCB 挂到该控制器的等待队列上；若空闲，则将控制器分配给进程。
- 3) 分配通道。在该 COCT 中又可找到与该控制器连接的通道的 CHCT，再根据 CHCT 内的状态信息，可知该通道是否忙碌。若忙，则将请求 I/O 的进程挂到该通道的等待队列上；若空闲，则将该通道分配给进程。只有在上述三者都分配成功时，这次设备的分配才算成功。然后，便可启动该 I/O 设备进行数据传送。

为使独占设备的分配具有更强的灵活性，提高分配的成功率，还可从以下两方面对基本的设备分配程序加以改进：

- 1) 增加设备的独立性。进程使用逻辑设备名请求 I/O。这样，系统首先从 SDT 中找出第一个该类设备的 DCT。若该设备忙，则又查找第二个该类设备的 DCT。仅当所有该类设备都忙时，才把进程挂到该类设备的等待队列上；只要有一个该类设备可用，系统便进一步计算分配该设备的安全性。
- 2) 考虑多通路情况。为防止 I/O 系统的“瓶颈”现象，通常采用多通路的 I/O 系统结构。此时对控制器和通道的分配同样要经过几次反复，即若设备（控制器）所连接的第一个控制器（通道）忙时，则应查看其所连接的第二个控制器（通道），仅当所有控制器（通道）都忙时，此次的控制器（通道）分配才算失败，才把进程挂到控制器（通道）的等待队列上。而只要有一个控制器（通道）可用，系统便可将它分配给进程。

设备分配过程中，先后分别访问的数据结构为 SDT→DCT→COCT→CHCT。要成功分配一个设备，必须要：① 设备可用；② 控制器可用；③ 通道可用。所以，“设备分配，要过三关”。

## 参 考 文 献

- [1] 汤子瀛. 计算机操作系统[M]. 西安: 西安电子科技大学出版社, 2001.
- [2] 李善平. 操作系统学习指导和考试指导[M]. 杭州: 浙江大学出版社, 2004.
- [3] William Stallings. 操作系统: 精髓与设计原理[M]. 北京: 机械工业出版社, 2010.
- [4] Tanenbaum A. S. 现代操作系统[M]. 北京: 机械工业出版社, 2009.
- [5] 本书编写组. 计算机专业基础综合考试大纲解析[M]. 北京: 高等教育出版社, 2009.
- [6] 李春葆等. 操作系统统考辅导教程[M]. 北京: 清华大学出版社, 2010.
- [7] 崔魏等. 计算机学科专业基础综合辅导讲义[M]. 北京: 原子能出版社, 2011.
- [8] 翔高教育. 计算机学科专业基础综合复习指南[M]. 上海: 复旦大学出版社, 2009.
- [9] Bryant R. E. 等. 深入理解计算机系统[M]. 北京: 机械工业出版社, 2010.