

程序“拥有”该扩展名。当用户双击某个文件名时，“拥有”该文件扩展名的程序就启动并运行该文件。例如，双击file.doc启动了Microsoft Word程序，并以file.doc作为待编辑的初始文件。

#### 4.1.2 文件结构

文件可以有多种构造方式，在图4-2中列出了常用的三种方式。图4-2a中的文件是一种无结构的字节序列，操作系统事实上不知道也不关心文件内容是什么，操作系统所见到的就是字节，其任何含义只在用户程序中解释。在UNIX和Windows中都采用这种方法。

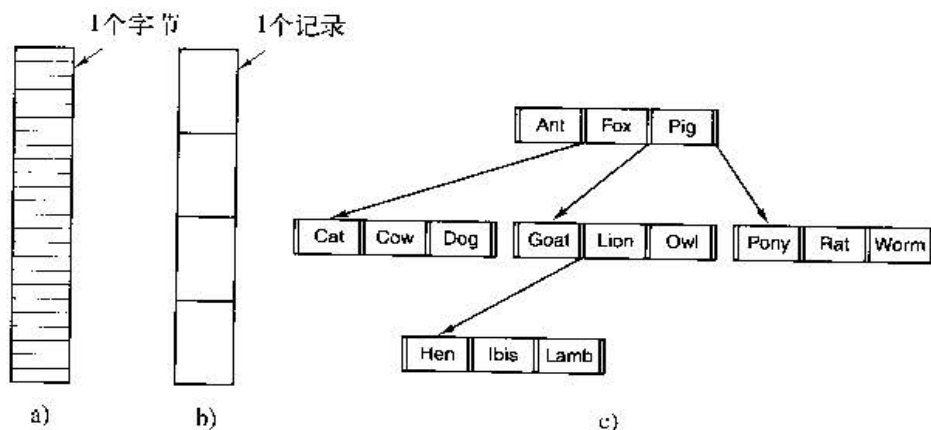


图4-2 三种文件结构：a) 字节序列；b) 记录序列；c) 树

把文件看成字节序列为操作系统提供了最大的灵活性。用户程序可以向文件中加入任何内容，并以任何方便的形式命名。操作系统不提供任何帮助，但也不会构成阻碍。对于想做特殊操作的用户来说，后者是非常重要的。所有UNIX、MS-DOS以及Windows都采用这种文件模型。

图4-2b表示在文件结构上的第一步改进。在这个模型中，文件是具有固定长度记录的序列，每个记录都有其内部结构。把文件作为记录序列的中心思想是：读操作返回一个记录，而写操作重写或追加一个记录。这里对“记录”给予一个历史上的说明，几十年前，当80列的穿孔卡片还是主流的时候，很多（大型机）操作系统把文件系统建立在由80个字符的记录组成的文件基础之上。这些操作系统也支持132个字符的记录组成的文件，这是为了适应行式打印机（当时的行式打印机有132列宽）。程序以80个字符为单位读入数据，并以132个字符为单位写数据，其中后面52个字符都是空格。现在已经没有以这种方式工作的通用系统了，但是在80列穿孔卡片和132列宽行式打印机流行的日子里，这是大型计算机系统常见模式。

第三种文件结构如图4-2c所示。文件在这种结构中由一棵记录树构成，每个记录并不具有同样的长度，而记录的固定位置上有一个“键”字段。这棵树按“键”字段进行排序，从而可以对特定“键”进行快速查找。

虽然在这类结构中取“下一个”记录是可以的，但是基本操作并不是取“下一个”记录，而是获得具有特定键的记录。如图4-2c中的文件zoo，用户可以要求系统取键为pony的记录，而不必关心记录在文件中的确切位置。进而，可以在文件中添加新记录。但是，把记录加在文件的什么位置是由操作系统而不是用户决定的。这类文件结构与UNIX和Windows中采用的无结构字节流明显不同，但它在一些处理商业数据的大型计算机中获得广泛使用。

#### 4.1.3 文件类型

很多操作系统支持多种文件类型。如UNIX和Windows中都有普通文件和目录，UNIX还有字符特殊文件（character special file）和块特殊文件（block special file）。普通文件（regular file）中包含有用户信息。图4-2中的所有文件都是普通文件。目录（directory）是管理文件系统结构的系统文件，将在以后的章节中讨论。字符特殊文件和输入/输出有关，用于串行I/O类设备，如终端、打印机、网络等。块特殊文件用于磁盘类设备。本章主要讨论普通文件。

普通文件一般分为ASCII文件和二进制文件。ASCII文件由多行正文组成。在某些系统中，每行用