

文件。通过这种方法，不需要任何特殊的机制就能进行I/O。

特殊文件（设备）分为两类，块特殊文件和字符特殊文件。一个块特殊文件由一组具有编号的块组成。块特殊文件的主要特性是：每一个块都能够被独立地寻址和访问。也就是说，一个程序能够打开一个块特殊文件，并且不用读第0块到第123块就能够读第124块。磁盘就是块特殊文件的典型应用。

字符特殊文件通常用于表示输入和输出字符流的设备。键盘、打印机、网络、鼠标、绘图机以及大部分接受用户数据或向用户输出数据的设备都使用字符特殊文件来表示。访问一个鼠标的124块是不可能的（甚至是无意义的）。

每个特殊文件都和一个处理其对应设备的设备驱动相关联。每个驱动程序都通过一个主设备号来标识。如果一个驱动程序支持多个设备，如，相同类型的两个磁盘，每个磁盘使用一个次设备号来标识。主设备号和次设备号结合在一起能够惟一地确定每个I/O设备。在很少的情况下，一个单独的驱动程序处理两种关系密切的设备。比如：与“/dev/tty”联合的驱动程序同时控制着键盘和显示器，这两种设备通常被认为是一种设备，即终端。

大部分的字符特殊文件都不能够被随机访问，因此它们通常需要用不同于块特殊文件的方式来控制。比如，由键盘上键入输入字符并显示在显示器上。当一个用户键入了一个错误的字符，并且想取消键入的最后一个字符时，他敲击其他的键。有人喜欢使用“backspace”回退键，也有人喜欢“del”删除键。类似地，为了取消刚键入的一行字符，也有很多方法。传统的方法是输入“@”，但是随着e-mail的传播（在电子邮件地址中使用@），一些系统使用“CTRL+U”或者其他字符来达到目的。同样的，为了中断正在运行的程序，需要使用一些特殊的键。不同的人有不同的偏爱。“CTRL+C”是常用的方法，但不是惟一的。

Linux允许用户自定义这些特殊的功能，而不是强迫每个人使用系统选择的那种。Linux提供了一个专门的系统调用来设置这些选项。这个系统调用也处理tab扩展，字符输出有效、失效，回车和换行之间的转换等类似的功能。这个系统调用不能用于普通文件和块特殊文件。

10.5.2 网络

I/O的另外一个例子是网络，由Berkeley UNIX首创并在Linux中差不多原封不动引入。在Berkeley的设计中，关键概念是套接字（socket）。套接字与邮筒和墙壁上的电话插座是类似的，因为套接字允许用户连接到网络，正如邮筒允许用户连接到邮政系统，墙壁上的电话插座允许用户插入电话并且连接到电话系统。套接字的位置见图10-19。套接字可以被动态创建和销毁。创建一个套接字成功后，系统返回一个文件描述符。创建连接、读数据、写数据、解除连接时要用到这个文件描述符。

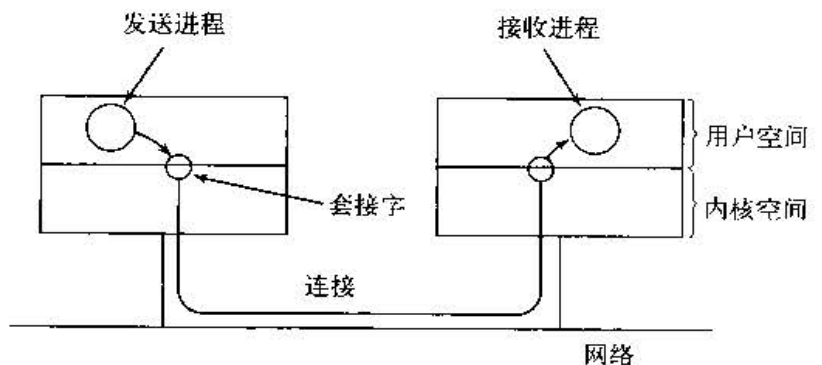


图10-19 网络中使用套接字

每个套接字支持一种特定的网络类型，这在套接字创建时指定。最常用的类型是：

- 1) 可靠的面向连接的字节流。
- 2) 可靠的面向连接的数据包流。
- 3) 不可靠的数据包传输。

第一种套接字类型允许在不同机器上的两个进程之间建立一个等同于管道的连接。字节从一个端点注入然后按注入的顺序从另外一个端点流出。系统保证所有被传送的字节都能够到达，并且按照发送时的顺序到达。

除保留了数据包之间的分界之外，第二种类型和第一种是相同的。如果发送者调用了5次写操作，每次写了512字节，而接收者要接收2560字节，那么使用第一种类型的套接字，接收者接收一次会立刻接收到所有2560个字节。要是使用第二种类型的套接字，接收者一次只能收到512个字节，而要得到剩