

- 改变当前的文件位置。对于每个打开的文件，内核保持着一个文件位置 k ，初始为 0。这个文件位置是从文件开头起始的字节偏移量。应用程序能够通过执行 seek 操作，显式地设置文件的当前位置为 k 。
- 读写文件。一个读操作就是从文件复制 $n > 0$ 个字节到内存，从当前文件位置 k 开始，然后将 k 增加到 $k + n$ 。给定一个大小为 m 字节的文件，当 $k \geq m$ 时执行读操作会触发一个称为 end-of-file(EOF)的条件，应用程序能检测到这个条件。在文件结尾处并没有明确的“EOF 符号”。

类似地，写操作就是从内存复制 $n > 0$ 个字节到一个文件，从当前文件位置 k 开始，然后更新 k 。

- 关闭文件。当应用完成了对文件的访问之后，它就通知内核关闭这个文件。作为响应，内核释放文件打开时创建的数据结构，并将这个描述符恢复到可用的描述符池中。无论一个进程因为何种原因终止时，内核都会关闭所有打开的文件并释放它们的内存资源。

10.2 文件

每个 Linux 文件都有一个类型(type)来表明它在系统中的角色：

- 普通文件(regular file)包含任意数据。应用程序常常要区分文本文件(text file)和二进制文件(binary file)，文本文件是只含有 ASCII 或 Unicode 字符的普通文件；二进制文件是所有其他的文件。对内核而言，文本文件和二进制文件没有区别。

Linux 文本文件包含了一个文本行(text line)序列，其中每一行都是一个字符序列，以一个新行符(“\n”)结束。新行符与 ASCII 的换行符(LF)是一样的，其数字值为 0x0a。

- 目录(directory)是包含一组链接(link)的文件，其中每个链接都将一个文件名(filename)映射到一个文件，这个文件可能是另一个目录。每个目录至少含有两个条目：“.”是到该目录自身的链接，以及“..”是到目录层次结构(见下文)中父目录(parent directory)的链接。你可以用 mkdir 命令创建一个目录，用 ls 查看其内容，用 rmdir 删除该目录。
- 套接字(socket)是用来与另一个进程进行跨网络通信的文件(11.4 节)。

其他文件类型包含命名通道(named pipe)、符号链接(symbolic link)，以及字符和块设备(character and block device)，这些不在本书的讨论范畴。

Linux 内核将所有文件都组织成一个目录层次结构(directory hierarchy)，由名为/(斜杠)的根目录确定。系统中的每个文件都是根目录的直接或间接的后代。图 10-1 显示了 Linux 系统的目录层次结构的一部分。

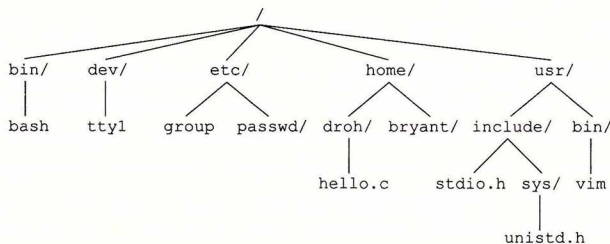


图 10-1 Linux 目录层次的一部分。尾部有斜杠表示是目录