

5) 系统管理。

6) 其他。

POSIX 1003.2标准规定了100种左右关于上述程序的语法和语义，主要是前三类中的程序。让这些程序具有统一的标准主要是为了实现让任何人写的shell脚本可以在任何Linux系统上运行。

除了这些标准应用程序外，当然还有许多其他应用程序，比如Web浏览器，图片浏览器等。

下面我们看一看一些程序的例子，首先从文件和目录操作开始。

`cp a b`

将文件a移动到b，而不改变原文件。相比之下，

`mv a b`

将文件a移动到b但是删除原文件。从效果上来看，它是文件移动而不是通常意义上的复制。cat命令可以把多个文件的内容连接起来，它读入每一个输入文件然后把它们按顺序复制到标准输出中。可以通过rm命令来删除文件。命令chmod可以让属主通过修改文件的权限位来改变其访问权限。使用mkdir和rmdir命令可以分别实现目录的创建和删除。为了列出一个目录下的文件，可以使用ls命令。它包含大量的标志来控制要显示文件的哪些特征（如大小、用户、群、创建日期）、决定文件的显示顺序（如字母序、修改日期、逆序）、指定文件输出格式等。

我们已经见到了很多过滤器：grep从标准输入或者一个或多个输入文件中提取包含特定模式的行；sort将输入进行排序并输出到标准输出；head提取输入的前几行；tail提取输入的后几行。其他的由1003.2定义的过滤器有：cut和paste，它们实现一段文档的剪切和粘贴；od将输入（通常是二进制）转换成ASCII文档，包括八进制，十进制或者十六进制；tr实现字符大小写转换（如小写换大写），pr为打印机格式化输出，包括一些格式选项，如运行头，页码等。

编译器和程序设计工具包括gcc（它调用C语言编译器）以及ar（它将库函数收集到存档文件中）。

另外一个重要的工具是make，它负责维护大的程序，这些程序的源码通常分布在多个文件中。通常，其中一些文件是头文件（header file），其中包括类型、变量、宏和其他声明。源文件通常使用include将头文件包含进来。这样，两个或更多的源文件可以共享同样的声明。然而，如果头文件被修改，就需要找到所有依赖于这个头文件的源文件并对它们重新进行编译。make的作用是跟踪哪些文件依赖于哪些头文件等，然后安排所有需要进行的编译自动进行。几乎所有的Linux程序，除了最小的那些，都是依靠make进行编译的。

一部分POSIX标准应用程序列在图10-2中，包括每个程序的简要说明。所有Linux系统中都有这些程序以及许多其他标准的应用程序。

程 序	典 型 应 用
cat	将多个文件连接到标准输出
chmod	修改文件保护模式
cp	复制一个或多个文件
cut	从一个文件中剪切一段文字
grep	在文件中检索给定模式
head	提取文件的前几行
ls	列出目录
make	编译文件生成二进制文件
mkdir	创建目录
od	以八进制显示一个文件
paste	将一段文字粘贴到一个文件中
pr	为打印格式化文件
ps	列出正在运行的进程
rm	删除一个或多个文件
rmdir	删除一个目录
sort	对文件中的所有行按照字母序进行排序
tail	提取文件的最后几行
tr	在字符集之间转换

图10-2 POSIX定义的一些常见的Linux应用程序

10.2.5 内核结构

在图10-1中我们看到了Linux系统的总体结构。在进一步研究内核的组成部分，如进程调度和文件系统之前，我们先从整体的角度看一下Linux的内核。

内核坐落在硬件之上，负责实现与I/O设备和存储管理单元的交互，并控制CPU对前述设备的访问。如图10-3所示，在最底层，内核包含中断处理程序，它们是与设备交互的主要方式，以及底层的分派机制。这种分派在中断时发生。底层的代码中止正在运行的进程，将其状态存储在内核进程结构中，然后