UNIX
程序开发环境

Brian W.Kernighan
Rob Pike 著
平镇中 何爱芳 编译
杨乔林 校

《UNIX程序开发环境》编辑部
1985

编 译 前 言

UNIX现已成为应用得最广泛的分时操作系统，从小型微计算机到巨型计算机数以万计的UNIX系统在运转。

UNIX操作系统之所以成功，不仅因为是由高效的、便于移植的C语言所编写的，这一特性使得UNIX系统可以不断改进、可以适应各种硬件环境；而更重要的是在发展过程中形成了一个丰富灵活的运行环境，这一环境为程序设计提供了广泛的资源，很多问题的求解可以不必按传统方法编程，而仅需把现有的资源组织在一起。UNIX许多命令看起来简单而普通，但是当它们组织在一起时就表现出强有力的功能和广泛的用途。UNIX系统之所以有效，就在于它的新的程序设计原理和新的使用计算机方法。

本书着重介绍UNIX的新程序设计原理和如何使用UNIX编程环境的资源，它通过大量的程序实例，由浅入深地逐步引导读者掌握UNIX编程原理。本书第一、二章讲述UNIX系统基本知识，包括UNIX入门知识和UNIX的核心文件系统。第三章和第五章讲Shell程序设计。第四章讲UNIX的一类应用程序——滤通程序。第六、第七章介绍UNIX输入输出和系统调用，即UNIX系统C语言程序设计。第八、第九章讲进一步的UNIX程序开发和应用，包括Yacc程序的应用和资料编排工具。

本书的作者是Brian W. Kernighan和Rob Pike，他们在UNIX操作系统和C语言方面写有大量专著，他们在UNIX领域中享有的盛名是不用介绍的。

本书是一本研究UNIX系统的专门人员和UNIX用户的必备之书，本书选材适当、讲解深入浅出，不失为一本学习UNIX系统的极好教材。

在本书的编译过程中，杨乔林副研究员给予了很多指导，在此谨致谢意。另外在编译中对原文的内容略有删节和改动，以便更适合我国广大读者使用。最后由于编译时间仓促，文中难免有不当之处，望各位专家同行指教。

一九八五年九月

目　录

第一章　UNIX入门

什么是“UNIX”？狭义地说，它是指这个分时操作系统的内核，也就是管理计算机资源并为用户分配资源的程序。UNIX系统协助用户运行程序；UNIX系统管理计算机的外围设备，控制磁盘、终端和打印机的运行；UNIX系统提供了文件管理系统，管理外存中程序、数据和文件的存取。

广义地说，“UNIX”不仅包括系统的内核，同时还包括一系列基础程序，如编译程序，编辑程序，命令语言，文件的复制和打印程序等等。

更广泛地，“UNIX”甚至还包括用户开发的程序，如资料排印工具，统计分析程序和图形软件包。

对“UNIX”一词的解释，究竟哪种说法更为确切取决于所考虑的系统。我们在本书中所用的“UNIX”，其含义根据上下文而定。

UNIX系统有时看起来似乎很难使用，初学者往往不知道如何使用系统提供的工具。实际上，UNIX系统的入门并不困难，只要知道几个程序就可以开始工作。本章的目的在于帮助初学者尽快地熟悉本操作系统。本章所叙述的内容并不是详尽无遗的，而是一个概观。以后的章节将更详细地阐述这些内容。本章主要内容是：

- 基本操作—注册和注销，简单命令，改正输入错误，邮件，终端间通信。
- 常用操作—文件与文件系统，打印文件，目录，常用命令。
- 命令解释程序和Shell命令—文件名的缩写，输入输出重新定向，管道，删字符和删行符的设置，定义命令的搜索路径。

假如读者以前使用过UNIX系统，对本章内容都很熟悉，可以直接从第二章开始。

阅读本书时需要准备一份《UNIX程序员手册》，即使只阅读第一章；在必要时我们将告诉读者查阅手册有关部分，而避免复述手册的内容。本书的目的不是要代替手册，而是引导读者掌握怎样更好地使用手册。另外，书中所述内容可能与你的具体系统有所不同。你可以通过用户手册中的目录索引，查找具体系统的命令。用户应该学会熟练地使用手册。

最后劝告读者，不要害怕使用机器。对于初学者，不太可能发生损害自己和他人的事故。本章篇幅很长，最好的方法是阅读几节后就上机操作，从而得到透彻的了解。

1.1 起步

终端和键盘的基本约定

UNIX系统是以全双工的方式工作，即从键盘把字符输入系统，系统再将字符回送到终端，并显示出来。通常，回送到终端的字符与输入字符相同，因此操作员看到的正是自己输入的字符。但有些时候，系统不回送符号。例如，用户打入保密口令时，屏幕就没有显示。

键盘上大多数字符是普通打印字符，它们没有特殊含义。只有少数特殊字符指示计算机做专门的操作。其中最常见的特殊字符是回车键RETURN，它表示输入行结束；系统收到回车信息便认为输入的当前行结束，系统的响应是让屏幕游标回到下一行行首。

回车符是控制符的一个例子。控制符是指控制终端工作方式的非显示字符。在一般终端上，回车键均占有一个键位，不过大多数控制符没有单独相应的键。所以输入一般控制符必须先按下控制键，或称作CTL、CNTL、CTRL键，然后再按所对应的字符键。例如，输入回车符可以直接按回车键，也可以先按控制键，再按m键。CONTROL-m或CTL-m也是回车符。一些常用的控制符有：CTL-d，它表示终端的输入结束；CTL-g控制终端响铃；CTL-h称为退格键，用于改正输入的错误；CTL-i常称作Tab键，其功能是移动游标到下一个Tab位置，与标准打字机的Tab键类似。在UNIX系统里Tab键移动8个空格。在大部分终端上，退格和Tab有单独的键。

此外还有两个特殊键，一个是DELETE键，有时也称作REBOUT。另一个是BREAK键，有时也称作INTERRUPT键。大多数UNIX系统中，DELETE键表示立即终止程序。在有些系统里，也用CTL-c终止程序。一般说来，BREAK键与DELETE键、CTL-c的功能基本相同。

UNIX的一段会话

我们以操作员和系统对话作为实例，给出这些程序片断，有时还加上必要的解释。全书所有程序采用如下约定：操作员的输入用斜体字，计算机输出用打字机体，评注和解释也用斜体。下面是操作员与UNIX系统的一段会话：

```
login: you                                  键入注册名you
Password:                                   键入口令，系统不回送
You have mail.                              提醒用户有邮件
$                                           系统准备接收命令
$                                           重复键入回车符
$ date                                      询问时间
Sun Sep 25 23:02:57 EDT 1983
$ who                                       询问谁在使用机器
jlb      tty0     Sep 25  13:59
you      tty2     Sep 25  23:01
mary     tty4     Sep 25  19:03
doug     tty5     Sep 25  19:22
egb      tty7     Sep 25  17:17
bob      tty8     Sep 25  20:18
$ mail                                      阅读邮件
From doug Sun Sep 25 20:53 EDT 1983
give me a call sometime monday
?                                           按回车键，读下一信件
From mary Sun Sep 25 19:07 EDT 1983
Lunch at noon tomorrow?
? d                                         删除信件
$                                           信件读完
$ mail mary                                 写信给mary
lunch at 12 is fine                         信件结束
```

```
Ctl-d                    挂断终端
$                        会话结束
```

在这段会话中交换信件后就结束退出系统，通常还要做其它事情。下面便是些具体例子。

注册

UNIX系统规定每个用户必须有一个注册名和一个口令。注册名可从系统管理员处得到。UNIX系统能够连接各种各样的终端，系统通常使用小写字母。对于UNIX系统小写字母是至关重要，如果你的终端只有大写字母，这将给实际工作带来很多困难，最好还是另找一个合适的终端。

使用机器之前，应确保终端的开关设置恰当。终端开关的选择主要包括大小写、全双工、速度和波特率。接通终端要通过一些专门的方法，如拨通电话或某个专用的开关。在终端接通后，系统应显示：

```
login:
```

假若终端显示的是一些杂乱的符号，可能是由传输速率不匹配而引起。应该检查一下速度开关或其它开关的设置。若检查后仍不能正常工作，可以慢慢地敲击几下BREAK键或INTERRUPT键。若仍不出现Login，那么应该另寻帮助。

出现Login后，应该用小写字母输入注册名，并按RETURN键。如果使用保密口令，系统将提出询问，在输入口令时系统不回送保密口令。

注册一旦成功，系统便响应一个提示符，表明系统已准备接收用户命令，提示符经常是一个美元符号$ 或百分符号%。用户可以更改提示符号，具体方法将在以后介绍。提示符实际上是用户和系统的接口程序发出的，即命令解释程序Shell发出的。

系统响应提示符前，还可能显示一些例行信息。如通知用户有邮件，或者询问用户所使用的终端类型。系统将根据用户的回答设置终端的特征参数。

输入命令

终端显示提示符后，用户就可以输入命令请求系统执行。这里所谓命令就是请求调用某个程序。如当用户输入date命令时，系统就去调用date程序显示当前的日期和时间，终端屏幕上会出现类似下面的信息

```
$ date
Mon Sep 26 12:20:57 EDT 1983
$
```

当你输入命令时，一定不要忘记按回车键，因为系统只有收到回车键才认为命令行结束。

用户的下一命令可以是who，即询问当前有哪些用户挂在系统里。

```
$ who
rlm         tty0     Sep 26 11:17
pjw         tty4     Sep 26 11:30
gerard      tty7     Sep 26 10:27
mark        tty9     Sep 26 07:59
you         ttya     Sep 26 12:20
$
```

系统响应的第一列表示用户名，也就是用户的注册名，第二列表示终端名，即各用户使用的终端编号（tty为teletype的缩写），随后各列表示各用户注册登记的日期和时间。用户有兴趣的话可试一下命令：

```
$ who am i
you      ttya    Sep 26 12:20
$
```

这时系统仅回送用户自己的注册信息。假如用户打错了命令，系统将回答这一命令没有找到。

```
$ whom
whom: not found
$
```

当然，如果用户打错的命令恰好是系统的某一命令，系统就会执行该命令。

终端异常响应

有时用户终端会出现异常。例如，每个字符重复显示二次，或者按回车键后游标不移到下一行的行首。这是由于终端的状态设置错误或由状态突然紊乱引起的现象。通常关上终端电源并再次打开或者消去注册又重新注册便可恢复。UNIX系统设有stty命令，可以设置终端状态特性。在UNIX程序员手册stty（1）部分，便可找到设置终端特性的说明。假如用户使用的终端设有表格处理Tabs功能，可以键入如下命令，由系统代为处理：

```
$ stty-tabs
```

假如用户使用的终端具有设置表格Tabs的功能，命令Tabs将建立其正常功能：

```
$ tabs terminal-type
```

详见手册Tabs（1）命令部分。

敲错键盘的改正

当命令从键盘输入时，如果敲错了键，并且在按回车键前发现了错，则可用两种方法进行。第一，逐字修改。用删字符一次擦除一个符号，随即输入正确的符号。第二，整行修改。用删行符一次删除一行，然后从头重新输入命令。

删行符的省缺值为@，下面是命令行行首出现错误，删行后重新输入的例子：

```
$ddtae@                          命令敲错，删除整行
date
Mon Sep 26 12:23:39 EDT 1983
$
```

删字符的省缺值为#，每个删字符删除前一字符，直至行首（不能超越行首）。当某键敲错时，可立即进行改正。如：

```
$ dd#atte##e                     随时改正
Mon Sep 26 12:24:02 EDT 1983
$
```

删字符和删行符的省缺值因系统而异。在很多系统中用退格符（backspace）代替删

字符。用户可以按下面的方法校验系统所用的删字符。

```
$ datee←                          试用←
datee←: not found                 删字符不是←
$ datee#                          试用#
Mon Sep 26 12:26:08 EDT 1983      删字符是#
$
```

上面把退格符写成←，以便读者看到这一符号。另一个经常使用的删行符是CTL-u。

本文以@表示删行符，以#表示删字符，对不同的UNIX系统可能有不同的表示。系统的删字符和删行符可由用户自己定义，在环境剪裁部分将介绍如何由用户自己设置删除符。

当系统的程序或文件刚好要使用删字符、删行符时，可以前置反斜杠（backslash）＼符，以表示它们已不是删除符，而是符号本身。例如，要输入#符或@符时，可键入＼#或＼@。在输入＼@时系统可能将光标送到屏幕下一行行首，但这并不影响输入的结果，因为@已经记入系统缓冲区。

反斜杠有时也称作前导符（escape character），表示其后继符号作为一般符号。要删除一个反斜杠应键入二个删字符，即＼##。想一想这是为什么？

键盘输入的符号要经过一系列程序的检查和解释，符号最终作如何解释不仅取决于符号最终要送到哪，而且还取决于符号所经过的路径。除非关掉了回应（echo），否则键入的每个字符都立即回应，显示在终端上。在回车符输入之前，输入符号串暂时由系统内核保存起来，这样输入错误可由删字符或删行符改正。当删字符和删行符以反斜杠前导时，系统内核将不保留反斜杠，而只保留下一符号。当回车符键入后，内核将符号串送到命令解释程序处理。

练习1-1：解释下列命令行意味着什么？

```
$ date＼@
□
```

练习1-2：大多数UNIX系统（除UNIX系统第七版外）定义#号前导的符号串为注解行，命令解释程序跳过#号引导的行。在这一规则下，如果#号同时又为删字符，试解释下面这段程序运行结果。

```
$ date
Mon Sep 26 12:39:56 EDT 1983
$ #date
Mon Sep 26 12:40:21 EDT 1983
$ ＼#date
$ ＼＼#date
#date: not found
$
```

□

提前输入

UNIX系统内核随时接收来自键盘的讯息，即使内核正忙于处理其它事务也是如此。因此用户可以不断地键入，而不必等待系统的响应。假如系统正在进行终端输出，键入的字符和系统的输出显示在一起，键入的字符由系统单独存放，所以仍能得到正确的解释。

终止程序的运行

用户可根据具体系统的规定，可用INTERRUPT、DELETE或BREAK键来终止程序的运行。对于某些程序，如编辑程序ED，DELETE键只终止当前正在进行的编辑命令，并返回询问编辑命令的状态，而不退出编辑程序。关闭终端也能引起程序的终止。

当显示屏上输出信息太多太快，希望暂时停止输出，可以按CTL-S键。按CTL-S键之后，终端输出停止，用户程序也随之挂起，直至再按CTL-q键再次恢复输出为止。

注销

当系统回到询问命令状态时，按CTL-d键，这表示终端输入结束，用户工作结束。系统接收到终端的CTL-d键后，将完成注销的一系列例行操作，同时回送Login：到终端，等待下一个用户注册。有些系统仅以关闭终端实现注销。

信件

系统提供了用户之间通信的邮件系统，当用户打开终端注册登录时发现系统给出如下信息：

```
You have mail.
```

这时用户可过键入mail命令读取信件：

```
$ mail
```

mail程序将逐个显示用户的信件，并依照时间顺序，显示最新的信件。每显示一段信件，mail都询问用户是否要对该信件作些处理。若用户回答d，则表示删除信件；若仅按回车键，表示对信件不作任何改动（信件仍旧保存，下次还可读这一信件）；若回答p，则要求重复显示信件；s filename表示要把信件存入所命名的文件；若回答q，表示要从mail退出。

假如用户要发信件给另一用户nico，可以键入：

```
$ mail nico
Now type in text of the letter
on as many lines as you like…
After the last line of the letter
type a control-d
ctl-d.
$
```

CTL-d标志信件结束，即告诉mail该信件已输入完毕。假如用户在输入信件的过程中想中途退出mail，可以按DELETE键，这时已输入的部分信件存放在称为Dead.Letter的文件里，而不发送出去。

作为练习，可送信件给自己，然后键入mail读自己发的信件（发信给自己是一种设置

备忘录的方法）。

mail命令还有很多其它用法，例如发送事先准备好的信件，或一次送信给若干人。还可以用其它方法送信件。详见UNIX程序员手册mail（1）。mail（1）中的1表示手册第一部分中介绍的mail命令。手册第一部分是UNIX命令和实用程序的说明。如calendar（1）（日历表服务程序）等等。

用户间通话

当UNIX系统处于多用户的情况下，有时在终端上会突然显示出下述信息：

```
Message from mary tty7…
```

并伴随出现一阵嘟嘟响声。这是用户mary想和你通话而产生的信号。若你用如下命令响应她，

```
$ write mary
```

这就建立起了你和mary的通信线路，mary在她的终端上键入的内容同时显示在你的终端上，反之你键入的内容也显示在mary的终端上。为了区分终端上哪些是你输入的，哪些是mary输入的，我们使用如下通话协议：（o）表示一段话说完，并让对方发话，（oo）代表通话结束并退出程序。

```
Mary's terminal:                    Your terminal:
$ write you
                                    $ Message from mary tty7…
                                    write mary
Message from you ttya…
did you forget lunch? (o)
                                    did you forget lunch? (o)
                                    five@
                                    ten minutes (o)
ten minutes (o)
ok (oo)
                                    ok (oo)
                                    ctl-d
EOF
ctl-d
                                    $ EOF
$
```

除CTL-d键外，也可以使用DELETE退出Write命令。此外，你所输入的错误并不在mary的终端上显示出来。

如果不愿意别人干扰你的工作，可以使用mesg命令拒绝接收。当你向一个拒绝接收通话的用户发写命令、或者向没有注册的用户要求通话时，write命令会显示不能通话的原因。

新闻

很多UNIX系统提供新闻服务。其中包括通过电话线联结庞大的UNIX系统网：新闻网（netnews）和用户网（USENET）。新闻网并行地向各用户转播新闻，其命令为：

```
$ news
```

手册

UNIX程序员手册提供了大部分用户所需的系统使用的知识。第一部分是命令与实用程序。本章涉及的全部命令均在其中；第二部分是系统调用，以及本书的第六章、第七章论及的游戏程序的说明。其余各部分阐述C程序使用的函数、文件的格式以及系统的维护（这些部分的次序因UNIX系统版本不同而异。手册的开始列有索引，可通过索引迅速找到所需的命令。手册还介绍了UNIX系统工作的概况。

很多UNIX系统在盘中存储着一部联机使用的手册，以供用户在终端上查找。例如，查找who命令的用法，键入：

```
$ man who
```

man将who命令的用法显示在终端上，也可以通过man命令查找man的用法，只需键入：

```
$ man man
```

计算机辅助教学

系统提供命令learn，进行计算机辅助学习，帮助用户学习文件系统和基本命令，以及编辑程序，文件准备，C语言编程等的使用。当键入

```
$ learn
```

命令后系统将引导你学习上述内容，如没有learn可键入teach一试。

游戏

UNIX系统不仅有严肃的计算和事务处理软件，而且还有一些流行的游戏程序。详见手册第六部分。

1.2 常用操作：文件处理和常用命令

UNIX系统的信息存放在文件里，文件与普通的公务文件类似。每个文件有自己的名字、内容、存放地址以及其它一些管理信息，如：文件的用户，文件的大小等。文件可以是一封信，一个通讯录，或者是程序的源语句，程序的数据，甚至可以包括可执行的程序和其它非正文内容。

UNIX文件系统具有良好的结构。用户可以有效地保护自己的文件，免受他人干扰。UNIX系统提供了很多文件处理程序。在这里主要考察几个常用的文件处理程序，关于文件系统和与文件有关的各种命令将在第二章作详细介绍。

文件的生成—编辑程序

假如你想要键入一篇文章，一封信或一段程序，该怎样把这些信息存到计算机里呢？这些工作是由文本编辑程序完成，编辑程序存储和处理计算机的信息。几乎所有的UNIX系统都有屏幕编辑程序。屏幕编辑充分利用了计算机终端的功能，形象地显示了编辑过程文件内容的变化。最常见的二个屏幕编辑程序是vi和emacs。由于图形排版的困难和目前尚没有一个标准的屏幕编辑程序，在此我们不作详细的描述。

在你的系统里，UNIX传统的行编辑程序ed一定可以运行。ed是不考虑终端的特殊性

的，因而它几乎可以在所有的终端上运行。ed亦为很多基本程序的基础（包括一些屏幕编辑程序）。因此值得专门花些时间来讨论它。附录1是ed的简单说明。

不论使用什么样的编辑程序，都应能十分熟练地迅速地产生和编辑文件。这里以ed为例，讨论具体产生和编辑文件步骤。讨论的实例一定能在你的UNIX系统里运行，但不一定能在其它编辑程序上运行。

下面利用ed创建junk文件，该文件包含正文内容：

```
$ ed                                调用文本编辑程序
a                                   ed命令，附加正文
now type in
whatever text you want…
.                                   键入‘.’，终止附加命令
w junk                              把正文写入junk文件
39                                  ed显示文件的字符数
q                                   退出ed
$
```

命令a（附加）指示ed开始收集正文。“.”表示附加正文结束，它必须打入在行首位置；注意不要忘记输入符号.。在正文结束符.输入之前，所有ed的其它命令都当作正文内容，而不作为命令解释。

w（写）命令存储所有正文内容；“w junk”把正文信息存放到文件junk中，文件名可以自己选定，这是junk表示文件内容无关紧要的意思。

文件存入后，ed回送文件的字符数目。在发w命令前，没有任何内容存到盘里。假如你在这时挂断终端，信息并没有存到指定的文件中（在编辑时挂断终端，键入的数据存到ed.hup文件里，以备下次调出继续编辑）。假如在编辑时，系统发生紊乱（因为软件或硬件的意外故障造成），编辑的文件仅仅包括上次写命令存入的部分。只有发w命令后，文件信息才能完全记录下来，这样键入下面命令，便可再次编辑文件。

```
$ ed junk
```

这样，又可对文件再次进行编辑，如改正拼写错误，修改词句，重新安排段落或做一些其它工作。当你编辑完文件，可用q（“quit”）命令退出编辑程序。

文件名列表

如下方式建立了文件junk和temp：

```
$ ed
a
To be or not to be
.
w junk
19
q
$ ed
a
That is the question.
```

```
.
w temp
22
q
$
```

ed程序回送的字符总数包括行结束符，后者有时称为换行符，这一符号是系统处理回车符RETURN所建立的符号。

ls命令列出文件的名字（注意不是文件的内容）：

```
$ ls
junk
temp
$
```

列出的文件名正是刚刚建立的二个文件（可能还有早已建立的文件）。文件名按字母顺序列出。

ls命令同大多数命令一样，可以用选择项改变省缺值。选择项在命令行中的命令名后面，通常由一个前导负号‘-’加上单个字符来表示各种含义。例如，ls-t表示文件按时间顺序列出。即文件最后一次修改的时间排列，时间上修改得越靠后，顺序上排列得越靠前。

```
$ ls -t
temp
junk
$
```

选择项-l给出较长的列表形式，对每个文件提供更多的信息：

```
$ ls -l
total 2
-rw-r--r-- 1 you                    19 Sep 26 16:25 junk
-rw-r--r-- 1 you                    22 Sep 26 16:26 temp
$
```

“total2”表示文件总共占用盘空间2块；一般一块表示512字节或1024字节。字符串-rw-r—r—指出文件的访问许可权；这表示文件所属者（you）可以读写，而其它用户只能读不能写。随后的“1”表示文件的连接数；在第二章中我们再说明它的含义。“you”是文件主，即文件的建立者。19和22是文件的字符数，它等于ed写入文件后回送的字符数。日期和时间指出文件最后一次修改的时间。

选择项可以组合起来。ls-lt给出的信息同ls-l一样，不过文件的顺序是最后修改的排在前面。选择项-u给出文件使用时间信息，ls-lut给出（-l）列表形式，而且最晚用过的排在前面。选择项-r倒转列表的次序，因此ls-lrt将最早用过的文件排在最前面。对于甚感兴趣的文件，ls也可以列出有关的信息：

```
$ ls -l junk
-rw-r--r-- 1 you                    19 Sep 26 16:25 junk
$
```

命令行中命令后面的字符串便是程序参数，上例中的-l和junk便是。参数一般是命令

所要求的选择项和文件名。

按照通常的惯例，选择项由负号和单个字母组成，如-t或组合形式-lt。一般来说，如果命令接收选择项参数，这些参数应放在文件名参数之前，当然它们也可能以其它次序出现。UNIX系统对多个选择项没有一项确定的规则，例如，标准的UNIX 第七版不接受如下ls命令：

```
$ ls -l -t
```

应该写作ls-lt，但对于其它程序则要求多重选择项分开来写。

随着深入学习，你会明白选择项参数的写法有一定的规则可循。每个命令有其独具的功能，对于每个字母有特定的解释(同一功能对于不同的命令参数往往不同)。这些不一致的选择项使系统显得很不协调，往往看作系统的一个主要弊病。这一状况正在不断改善，新的版本采用愈趋一致的选择项。在这里我们所能做的只是建议：当你想更好地写程序时，应在身旁放一本手册供随时查询。

打印文件—cat和pr

现在你有了一些文件，怎样去查看它们的内容呢？能实现这一工作的程序很多，利用编辑程序是一种可行的方法。

```
$ ed iunk
19                          ed表示iunk有19个字符
1,$p                        打印第一行至最后一行
To be or not to be          文件只有一行
q                           全部做完，退出ed
$
```

ed开始报告文件junk的字符总数；命令1，$p表示打印整个文件。当你学会使用编辑程序后，可以只打印文件的一部分内容。

使用编辑程序打印文件往往受到一些限制：ed不能打印很大的文件，最多只能数千行；另外，一次也只能打印一个文件，而往往需要打印几个文件，一个接一个地不间断地打印。下面我们再提出二个打印程序供大家选择。

第一个是cat。cat是所有打印命令中最简单的一个。cat打印文件内容，文件名作为cat命令参数：

```
$ cat junk
To be or not to be
$ cat temp
That is the question.
$ cat junk temp
To be or not to be
That is the question.
$
```

在打印多数文件时，文件一个接一个地送往终端，文件之间没有插入其它符号。

如果显示的文件很长，可用ctl-s暂停cat的输出，以免信息上滚离开屏幕。UNIX没有帧显示的标准命令，有些系统称之为pg命令，而本书使用的是p命令。第六章将介绍页面显示的p命令的实现方法。

和cat一样，pr打印参数也是需要打印文件名。pr使用行打印机的打印格式：每页66行（11英寸）长，每页顶部还打印刊头，包括文件最后修改的日期和时间、页面序号及文件名。页和页之间，有若干行空行，以便跳过页间走纸间隙。在下例中，先打印junk文件，然后跳到新的一页顶部，继续打印temp文件。

```
$ pr junk temp
Sep 26 16:25 1983 junk page 1
To be or not to be
        (60 more blank lines)                    (60余行空行)
Sep 26 16:26 1983 temp page 1
That is the question.
        (60 more blank lines)                    (60余行空行)
$
```

pr还提供多列打印的输出格式：

```
$ pr -3 filenames
```

表示以3列的格式打印文件。也可用其他的列数代替3，而使pr打印得更好看（filenames是文件名表）。pr-m以并列格式打印若干个文件，详见pr（1）。

必须指出，pr不是一个排版程序，排版程序至少应该重新安排行列，重新调整边缘空白的数目。UNIX的排版程序有nroff和troff，它们将在第九章介绍。

还有使用高速打印机打印文件的命令。参阅手册里lp、lpr命令，或查索引中的printer命令。究竟使用什么命令取决于系统所接的设备。pr和lpr经常一起使用，pr设置适当的打印格式，lpr处理送往行打印机的输出。在以后的章节还要进一步讨论。

换名、复制与删除文件—mv，cp，rm

下面我们逐个考察这些命令：第一个是换名。这是通过改变文件的名字进行的。如：

```
$ mv junk precious
```

这一命令的作用是将原先称为junk的文件赋予新的名字precious；但文件内容不变。当再次运行ls命令，precious文件代替了junk文件。

```
$ ls
precious
temp
$ cat junk
can:ca n't open junk
$
```

必须注意，如果把一个文件换名为一个已经存在的文件名，那么后者内容就不再存在了。

复制文件（使一个文件有二个文本）可使用cp命令：

```
$ cp precious precious.save
```

这样，为precious拷贝了一个副本文件precious.save。

最后，当你不需要某个文件时，可用rm命令删除之。

当要删除的文件不存在时，你会得到一个出错提示信息；当文件存在时，大多数

UNIX系统将默默地执行命令。许多UNIX命令没有过多的提示和回应，错误信息也相当简单，似乎没有什么帮助。这种情况对新的用户不大习惯，而对有经验的用户来说，过多的命令提示是不必要的。

什么是文件名

到目前为止，我们使用了一些文件名但还未交待哪些是合乎要求的文件名。现在给出文件命名的二个规则。首先，文件名的长度要限制在14个符号以内。第二，尽管文件名几乎可用所有的符号，但一般来说文件名应该采用可见符号，并应避免使用具有其它含义的字符串。例如，在ls命令中，ls-t 意味着按时间次序列出文件名。所以假使你把一个文件叫作-t，在要列该文件时，将得到按时间顺序排列的很多文件名。除了负号，还有少数几个符号在作为第一个符号时，具有特殊的意义。为了避免这类命名引起的错误，最好使用字母，数字，句点和下划线（一般习惯用句点和下划线表示把文件名字段分隔，如上述的文件名precious.save）。最后不要忘记大小写的区别—junk，Junk和JUNK是三个不同的名字。

一些特殊用途的命令

我们已初步讨论了文件的建立，文件名列表以及打印文件内容的方法，也提到了一部分文件处理命令。为了使讨论更具体，我们以poem文件作例子，它是Augustus De Morgan写的诗。首先用ed建立文件poem：

```
$ ed
a
Great fleas have little fleas
    upon their backs to bite 'em,
And little fleas have lesser fleas,
    and so ad infinitum.
And the great fleas themselves, in turn
    have greater fleas to go on;
while these again have greater still,
    and greater still, and so on.
.
w poem
263
q
$
```

第一个命令是计算一个或若干个文件的行数，词数和字符数；因为该命令具有字计数功能，故称为we（word-counting）。

```
$ wc poem
       8      46     263 poem
$
```

即poem文件有8行，46个词，263个字符。词的定义很简单，不包括空白、tab和换行符的任何符号串就称之为词。

wc可以对多个文件进行计数（打印出总数），必要时可选择计数的项目，参见wc