

# UNIX操作系统第7版用户手册

UNIX入门和文件编制准备

第三卷 上集

中国科学院成都计算机应用研究所情报室译

UNIX操作系统用户手册

第三卷 UNIX入门和文件编制准备

## 目 录

### 第一章 了解 UNIX

- 1 • UNIX初学者入门——第二版..... ( 2 )
- 2 • UNIX分时系统..... (47)
- 3 • UNIX第7版概要..... (82)
- 4 • 与UNIX通信..... (120)
- 5 • UNIX命令一览表..... (160)

### 第二章 编 辑

- 1 • 编辑入门..... (164)
- 2 • 介绍UNIX正文编辑程序..... (210)
- 3 • UNIX的高级编辑..... (244)
- 4 • EX参考手册..... (298)
- 5 • 介绍用v1显示编辑..... (334)

### 第三章 编排正文格式和资料准备

- 1 • 利用一ms宏程序包编排UNIX上的资料格式..... (400)
- 2 • Troff辅导..... (440)

## 第一章 了解UNIX

### 1. UNIX初学者入门—第二版

Brian W. Kernighan (贝尔实验室)

#### 引言

从用户观点看来，UNIX是易学易用的，在完成作业中很少出现一般常有的障碍。不过对初学者来说，要知道从何入手和如何使系统功能得到充分利用也不容易。本文目的就是要帮助新用户掌握UNIX系统的主要思路，并迅速开始有效地使用它。

在阅读这份资料时，为了便于参考，你应当有几本别的资料。最重要的资料是“UNIX程序员手册”，在这本手册中阅读有些资料比重复阅读本文中的内容常常更为方便。另外一本有用的资料是后面的“UNIX正文编辑程序介绍”。它可以告诉你如何用编辑程序把正文（程序、数据和文本资料）放进计算机中。

提醒一下：UNIX系统已经成为十分普遍的系统，有几种重要的变种广泛使用。当然细节还在随时间改变。所以尽管所有UNIX的各种版本的基本结构和如何使用它是共同的，但是在你的系统上肯定有些东西与本文所介绍的不同。我们试图把这种问题减到最少，不过仍要了解有这种问题。如有疑问，要注意本文介绍的是7版的UNIX。

本文共分五节：

1. 启用：如何注册，如何打入，在打入时出了错误怎么办，如何注销。有些这类问题取决于你注册入那个系统（例如电话号码）和你用什么终端，所以这一节需要由用户所用系统的具体情况来补充。

2. 日常使用：为了有效地利用系统你每天需要的东西，一般常用的命令；文件系统。

3. 资料准备：准备原稿是 UNIX 系统最常用的用途之一。本节提出一些建议，但并没有提出有关广泛适用于任何那种格式化工具的指示。

4. 编写程序：UNIX 是一个极好的开发程序的系统。本节讲到一些工具，但并不是系统提供的程序设计语言的辅导材料。

## 一、启 用

### 注册

你一定得有一个 UNIX 的注册名，你可以从系统管理人那里取得这个注册名。如果你的系统不是使用常接终端，你还需要知道电话号码。UNIX 系统可以和各种各样的终端配合工作：

Terminet 300；Execuport，TI 和类似的手提式终端；像 HP 2640 之类的显示（CRT）终端等；像 Textronix 4014 那样的高价图形终端；像 GSI 和 DASI 那样的绘图终端，以及甚至各式各样的老式电传打字机。不过要注意：UNIX

只适合使用小写字母的设备。如果你的终端只能打出大写字母（例如 33 型电传打字机，某些显示终端和手提式终端），使用时就不大方便。最好你另外找一种终端使用。

在你的设备上一定要把切换开关放在适当位置上。可能需要调整的切换开关包括速度，大写字母/小写字母，全双工，偶校验及任何其它在具体设备上适用的开关。与你的终端建立联系的方法有两种：一种方法是拨电话机上的号，或者只搬动一下开关。不管用的是这两种办法中的那一种，UNIX 都应当在屏幕上给你打印出一个“login:”（注册）信息。如果 UNIX 给你打印出的符号乱七八糟，那么可能你的速度不对；就要检查切换开关的位置。如果还是不起作用，就要慢慢地按“break”键或“interrupt”键几次。如果还不能产生注册信息，就要向内行请教。

当出现 login: 信息时，就用小写字母打入你的注册名。接着打入一个 RETURN；在你打入 RETURN 之前，系统什么事也不干。如果要求有口令，系统就要问你要口令，当你打入口令时系统将关掉打印。不要忘记打 RETURN。

你的注册工作最后是出现一个“提示符”，提示符是一个单字符，它表示系统准备好接受你发出的命令。通常提示符是一个美元符号 \$ 或是一个百分数符号 %（在出现提示符之前，你也可能得到一个日期的信息，或者告诉你有邮件的通知）。

## 打入命令

一旦见到提示符，你就可以打入命令，也就是要系统于某件事的请求。你打入

```
date
```

接着再打RETURN。你应当得到以下信息

```
Mon Jan 16 14:17:10 EST 1978
```

在打入命令以后不要忘记打RETURN，否则系统无所反应。如果你觉得系统不理睬你，就打RETURN；系统应当出现一些信息。以后我们不再提到RETURN，不过不要忘记打入它。在每行末尾都要打RETURN。

另外一条你要试打的命令是who，它告诉你每个当前注册的人：

```
who
```

显示信息如下

```
mb      tty01    Jan16    09:11
ski     tty05    Jan16    09:33
gam     tty11    Jan16    13:07
```

这里的时间是用户注册的时间；“ttyxx”是用户所用终端的系统编号。

如果你打错了命令名，和错用一条并不存在的命令，系统就要通知你。例如你打入

`whom`

系统将通知你

`whom: not found` (找不到)

当然，如果你错打入某些别的命令名，系统仍然运行，但产生多少有点神秘难测的结果。

终端出现奇怪的动作

有时候你的终端会出现奇怪的动作。例如每个字母可能出现两次，或者打入 `RETURN` 但不换行，或者不返回到下一行的左端。这种问题通常可以用先注销然后再注册的办法来解决。或者你可以阅读本手册第一卷第一章中有关命令 `stty` 的介绍。如果你的终端上没有制表符 (`tab`)，为了要巧妙地处理制表符（在 `UNIX` 中这种制表符用得很多）可打入命令

`stty -tabs`

系统将为你把每个制表符转换成适当数目的空白。如果你的终端有计算机可设置的制表符，那么命令 `tab` 将为你正确地置停。

打入命令时的错误

如果你在打入命令时犯了错误，并在打入 `RETURN` 之前就发现了这个错误，有两个办法可以纠正。用字符 `#` 可以抹去新近打入的那个字符；事实上接连使用 `#` 字符可以倒回去抹掉直到这行开头为止的字符（但不超出这行）。所以如果你打得不好，你可以一边打入一



边校正：

```
dd #a t t e ## e
```

与直接打入 `date` 一样。

符号 `@` 可以抹去当前输入行上所有已打入的字符。所以如果这一行已经打得乱七八糟，不好校正，就可以打入一个 `@` 抹去这行，再从头开始。

如果你必须把 `#` 或 `@` 作为正文的一部分打入时怎么办呢？如果你在 `#` 或 `@` 的前面加上一条反斜线 `\`，它就失去它抹去字符的含义。所以要在某个正文中加入 `#` 或 `@` 时，可以打入 `\#` 或 `\@`。系统将在你打入 `@` 之后，即使在 `@` 的前面加有一条反斜线，也总是回答你一个换行。不用担心，`@` 已被记录下来了。

如果要抹去一条反斜线，就必须打入两个 `##` 或两个 `@@`，例如 `\##`。在 `UNIX` 中广泛利用反斜线来指示后面的那些字符有点特殊。

### 予读

`UNIX` 完全可以予读，它的意思是指你可以根据你的需要尽快打入。只要你需要，甚至在有些命令正打入给你时也没有关系，你能打多快就打多快。如果你在输出时打入，你输入的字符和输出的字符互相混杂出现，不过这两种字符将分别存贮，并按正确顺序解释。所以你可以一条接一条地打入几条命令，而不用等待第一条命令完成或

者甚至开始之后才打入第二条命令。

### 停止一个程序

打入字符“DEL”（在你的终端上可能叫做“delete”或“rubout”键）可以停止大部分的程序。也可以使用在大多数终端上找到的“interrupt”键或“break”键。在像正文编辑程序那样一些少数的程序中，按下DEL键可以停止正在运行的程序，但仍让你留在那个程序中。挂起电话可以停止大多数的程序。

### 注销

最容易的注销办法是把电话挂上。你也可以打入

```
l o g i n
```

并让别人使用你刚才用的那个终端。只断掉终端通常是不够的。大多数UNIX系不采用限时机制，所以除非你挂起电话，你将永远留在你的终端上。

### 邮寄

当你注册时，有时可能收到如下的信息

```
y o u h a v e m a i l （你有信件）。
```

UNIX可以提供一个邮递系统，所以你可以和系统的其它用户通信。如果要读出你的信件，可以打入命令

```
m a i l
```

你的信件就在屏上显示出来，一次显示一条消息，最新的消息首先显示。在每条消息显示出来以后，`mail`要等待你说要干什么。基本的回答是两种：`d`和`RETURN`。`d`删掉这条显示出来的消息，而`RETURN`并不删掉这条消息（所以下次你读你的信箱时这条消息还在那里）。在手册里还介绍了其它的回答。（有些旧的`mail`版本不是一次处理一条消息，但在其它方面相似）。

怎样向其它某个用户发送信件呢？假定你要把信寄给“`joe`”（假设“`joe`”是某个人的注册名）。最容易的办法就是：

```
mail joe
```

现在打入你的信的正文，打多少行随你喜欢…在信的最末一行之后打入字符“`control-d`”。也就是按下“`control`”键并打入字母“`d`”。`control-d`常叫做文件结束时的

“`EOF`”，在整个系统中用它来作为结束从终端输入的标记，所以你要熟悉它。

可以自己给自己发信当作练习。（这种作法初听起来可能有点奇怪，其实不然，给自己发信是一种提醒自己注意的方便的办法）。

还有别的发信办法，你可以发出一封早就准备好的信，你也可以一次给好多人发信。如果想了解更多细节可以看`mail(1)`。（记号`mail(1)`的意思是本手册第一卷第一章里的命令`mail(1)`）。

## 给其它用户写信

有时会意外地伴随着铃声收到以下信息

```
Message from joe tty07 .....
```

这个信息的意思是说Joe要和你谈话，不过除非你采取明显的动作，你并不能和他对讲。如果你要回答，可以打入命令

```
w r i t e   j o e
```

这条命令建立起一条双向的通信路径。现在不论Joe在他的终端上打入什么东西都会出现在你的终端上，而你在你的终端下不论打入什么东西也会出现在Joe的终端上。这条路径很慢，与和月球对话差不多。（如果你正在进行通信中，你必须进入能打入命令的状态。在常规情况下，不论你正在运行的是什么程序都必须结束或被结束。如果你正在编辑时，你可以临时从编辑程序中退出，可参阅编辑程序入门）。

为了防你打入的东西和Joe打入的东西互相混杂，需要有一个协议。典型的协议如下：

Joe打入write smith并且等待。

Smith打入write joe并且等待。

Joe现在打入他的消息（他喜欢打多少行就打多少行）。当他准备接收回答时，他打入(0)作为发出的信号，表示“完了”。

现在 Smith 打入一个回答，也用 (0) 结束。

这一过程循环往复，直到两人中的某个感到疲倦为止；以后他发出一个 (00) 信号表示他想退出，信号 (00) 的意思就是“完了并退出”。

为了结束对话，每方都必须单独在一行上打入“control-d”字符。（也可以用“delete”键）。当别的人打入他的“control-d”时，你可以在你的终端上收到消息 EOF。

如果你的信是写给某个没有注册的人，或者这个人不想有人干扰他。那么系统会把这个情况在屏上通知你。如果你写信的对象是注册过的，但他不回答你，在过了一会儿之后，只要打入“control-d”就行了。

### 联机手册

UNIX 程序员手册一般都保存在系统中。如果你对有些东西搞不清楚，而又找不到一个内行来帮助你，你可以在你的终端上打印那些可能对你有帮助的手册中的某些节。这种做法也可以用于获得有关某一条命令的最新信息。为了打印手册中的一节，可打入“man 命令名”。因此要读 who 命令时可打入

```
man who
```

当然，打入

man man

就会告诉你所有关于 man 命令的情况。

### 计算机辅助数学

你的 UNIX 系统可能有一个名叫 learn (学习) 的现成程序。它提供计算机辅助数学, 数学内容有关文件系统和基本命令、编辑程序、文件编制准备以及 C 语言。如果你的系统里有 learn 程序, 可以试打命令

learn

它将告诉你要干什么。

## 二、日常使用

### 建立文件—编辑程序

如果你必须打入一篇文章, 或一封信, 或一个程序, 那么你怎样才能把这些信息存贮在机器里去呢? 大部分这些任务都由 UNIX 的“正文编辑程序” ed 来完成。由于 ed 的有关资料完全编制在 ed (1) 中, 并且在后面的“UNIX 正文编辑程序介绍”一文中要作出详尽解释, 所以不在这里花时间来介绍如何使用它。目前我们要用它来作的全部工作就是要构成一些文件。(文件是一组存贮在机器里的信息, 这是一个简略的但是恰当的定义)。

为了建立一个名叫 junk 的文件(这个文件中有一些正文)

可以按以下步骤操作：

`ed junk` (调用正文编辑程序)

`a` (属于“`ed`”的命令，加上正文)

此后打入你想编入的任何正文…

• (表示加入正文结束的信号)

通知加上正文结束的“.”必须在新起一行的开头。不要忘记这一点，因为在你打入“.”之前，别的`ed`命令将都不能当作是`ed`命令，这时你打入的每样东西全被当作是加入的正文。

这时你可以在你打入的正文上进行各种各样的编辑操作，如像校正拼写错误，重排正文段落之类。最后你必须使用编辑程序的命令`w`把你已经打入的信息写入一个文件中去：

`w`

`ed`将在屏上显示它已写入文件`junk`中的字符数目。

在打入命令`w`之前，任何信息也没有长期保存，所以如果你挂上电话回家，信息就丢去了。但是在打入命令`w`以后，信息就长期保存起来，你只要打入

`ed junk`

就可以重新存取这些信息。打入一条`q`命令就可以退出编辑程序。

(如果你没有写入就试图退出，`ed`将在屏上显示一个?提醒你，

你只要第二次打入q就可以退出)。

现在按同样方式建立第二个名叫temp的文件。你现在应当有两个文件：junk和temp。

输出什么文件？

ls命令(用于“列表输出”)把UNIX所知道的任何文件的名称(不是内容)列表输出。如果你打入

```
ls
```

则回答是

```
junk
```

```
temp
```

这的确是刚才建立的那两个文件。文件的名称将自动地按照字母顺序排列，但也可能有别的变动。例如命令

```
ls -t
```

使文件按它们最近改动的顺序列表，最近的首先列出。选择-l任选项时可得到一个“长格式”的列表输出：

```
ls -l
```

将产生如下的输出

```
-rw-rw-rw- 1 bwk 41 Jul 22 2:56 junk  
-rw-rw-rw- 1 bwk 78 Jul 22 2:57 temp
```

日期和时间是最近一次修改文件的日期和时间。41和78是



字符数（这个数字应当和你从 `ed` 那里得到的数字一致）。`bwk` 是文件的主人，也就是建立文件的那个人。`-rw-rw-rw-` 告诉你谁有许可权读写文件，在目前情况下是任何人都许可读写。

任选项可以组合：`ls -lt` 的作用和 `ls -l` 一样，但按时间先后顺序排列。你还可以提出你感兴趣的那些文件的名字，而 `ls` 只列表输出有关它们的信息。在 `ls (l)` 中可以找到更多的资料。

使用以负号开头的任选变元，如像 `-t` 和 `-lt`，是 UNIX 程序一种常用的约定。一般情况下，如果一个程序采用了这种任选变元，那么这些变元都放在文件名变元之前。用空白隔开各个不同的变元也很重要，例如 `ls -l` 和 `ls -l` 是不同的。

### 打印文件

既然你已经有了一个正文文件，你怎么把它打印出来给人们观看呢？有许多程序都可以作这个工作，也许太多了。

一种简单的办法是利用编辑程序，因为在编辑时进行改动之前常常要打印。你可以打入命令

```
ed junk
```

```
l, sp
```

`ed` 将回答 `junk` 中的字符个数，然后打印这个文件中所有各行。在你学会如何使用编辑程序以后，你就可以选择你打印的部分。