

适用于 IBM PC AT 286 386 及其兼容机

SCOXENIX 系统 V

技术丛书

(共五册)

用户指南



2

上海电子计算机厂

中国科学院希望高级电脑技术公司

适用于 TBM PC AT 286 386 及其兼容机

汉字XENIX 系统V

技 术 丛 书

(共五册)

用 户 指 南

(二)

上海电子计算机厂

中国科学院希望高级电脑技术公司

一九九〇年九月

第一章 绪 言

§ 1.1 概述

§ 1.2 本手册介绍

§ 1.1 概述

本手册介绍几个基本的 XENIX 实用程序,包括 mail,正文编程序和强有力的开发环境 shell。

§ 1.2 本手册介绍

本手册是按如下组织的:

第一章“绪言”概述 XENIX 系统。

第二章“vi”介绍怎样使用正文编辑程序 vi(c)。

第三章“mail”讲实用程序 mail(c),并介绍怎样发送和接收邮件。

第四章“shell”,讲 shell 用法,命令说明和怎样写 shell 过程。

第五章“bc:计算器”介绍一个完善的计算器程序 bc(c)的使用方法。

第六章“建立通信系统”介绍怎样建立一个在 XENIX 和/或 UNIX 系统之间用拨号电话通信线实现通信的系统。

第七章“C-shell”介绍 csh(c)的使用。包括 C-shell 的语法及功能,命令及特点和怎样建立 shell 过程。

第八章“V-sh”,介绍全屏幕 shell 的用法及特点,它是一个菜单形式的 shell。本章是对已熟悉 XENIX 的一般概念的读者讲的,但初级用户也可以用 vsh。

附录 A“ed”介绍编辑程序 ed(c)的用法。

第二章

vi: 正文编辑程序

§ 2.1 绪言

§ 2.2 演示范例

§ 2.2.1 进入编辑程序

§ 2.2.2 插入正文

§ 2.2.3 命令的重复执行

§ 2.2.4 命令的撤消

§ 2.2.5 光标的移动

§ 2.2.6 删除

§ 2.2.7 模式检索

§ 2.2.8 检索和替代

§ 2.2.9 退出 vi(正文编辑)

§ 2.2.10 正文中插入其他文件

§ 2.2.11 暂时退出 vi

§ 2.2.12 改变显示方式

§ 2.2.13 取消编辑过程

§ 2.3 编辑任务

§ 2.3.1 如何进入编辑状态

§ 2.3.2 移动光标

§ 2.3.3 光标在文件中的移动;Scrolling

§ 2.3.4 在光标前插入内容;i 和 I 命令

§ 2.3.5 在光标后增加内容;a 和 A 命令

§ 2.3.6 修改键入错误

§ 2.3.7 开始新的一行

§ 2.3.8 重复上次插入内容

§ 2.3.9 将其它文件插入正文

§ 2.3.10 在正文中插入控制字符

§ 2.3.11 行与行间的连接和截断

§ 2.3.12 删除单个字符命令;x 和 X

§ 2.3.13 删除词命令:dw

§ 2.3.14 删除行命令:D 和 dd

§ 2.3.15 删除所有插入内容

§ 2.3.16 删除和替代正文

§ 2.3.17 正文的移动

§ 2.3.18 检索命令:/和?

§ 2.3.19 检索和替代

§ 2.3.20 模式匹配

§ 2.3.21 命令的取消

§ 2.3.22 命令的重复

第一章 绪言	(1)
§ 1.1 概述	(1)
§ 1.2 本手册介绍	(1)
第二章 vi:正文编辑程序	(2)
§ 2.1 简介	(2)
§ 2.2 演示范例	(2)
§ 2.3 编辑任务	(12)
§ 2.4 解决一些常见问题	(35)
§ 2.5 建立用户环境	(36)
§ 2.6 指令汇总	(38)
第三章 mail:邮件	(43)
§ 3.1 简介	(43)
§ 3.2 演示范例	(43)
§ 3.3 基本概念	(45)
§ 3.4 mail 的使用	(47)
§ 3.5 命令	(50)
§ 3.6 暂时退出编写方式	(55)
§ 3.7 通信环境的设置:mail 文件	(58)
§ 3.8 mail 的深入课题	(60)
§ 3.9 快速参考	(62)
第四章 shell	(68)
§ 4.1 简介	(68)
§ 4.2 基本概念	(68)
§ 4.3 shell 变元	(74)
§ 4.4 shell 状态	(78)
§ 4.5 命令的环境	(79)
§ 4.6 调用 shell	(80)
§ 4.7 shell 过程的参数传递	(81)
§ 4.8 控制 shell 的控制流程	(82)
§ 4.9 特殊的 shell 命令	(82)
§ 4.10 shell 过程的建立和组织	(94)
§ 4.11 执行状态标志	(95)
§ 4.12 支撑命令及其特征	(96)
§ 4.13 有效的 shell 程序设计	(101)
§ 4.14 shell 过程举例	(104)
§ 4.15 shell 文法	(111)
第五章 bc:计算器	(114)
§ 5.1 简介	(114)
§ 5.2 演示说明	(115)

§ 5.3 任务	(116)
§ 5.4 语文参考说明	(123)
第六章 建立通讯系统	(130)
§ 6.1 简介	(130)
§ 6.2 建立通讯系统需要做的工作和硬件支持	(131)
§ 6.3 安装一条直接的通讯线	(131)
§ 6.4 安装调制解调器	(133)
§ 6.5 安装 uucp	(136)
§ 6.6 系统维护	(148)
§ 6.7 操作细则	(150)
第七章 c-shell	(158)
§ 7.1 引论	(158)
§ 7.2 进入 c-shell	(158)
§ 7.3 使用 shell 变量	(159)
§ 7.4 使用 c-shell 历史表	(161)
§ 7.5 使用别名	(163)
§ 7.6 输入/输出重定向	(164)
§ 7.7 创建后台和前台作业	(164)
§ 7.8 使用内部命令	(165)
§ 7.9 创建命令程序	(166)
§ 7.10 使用参数变量	(166)
§ 7.11 替换 shell 变量	(166)
§ 7.12 使用表达式	(168)
§ 7.13 一个命令程序的实例	(168)
§ 7.14 使用其它控制结构	(170)
§ 7.15 为命令提供输入	(171)
§ 7.16 捕获中断	(171)
§ 7.17 使用其它特性	(172)
§ 7.18 在终端上使用循环	(172)
§ 7.19 使用花括号中的参数	(173)
§ 7.20 替换命令	(173)
§ 7.21 特殊字符	(173)
第八章 v-shell	(175)
§ 8.1 什么是 Visual shell	(175)
§ 8.2 visual shell 入门	(176)
§ 8.3 visual shell 屏幕	(176)
§ 8.4 visual shell 命令介绍	(179)
附录 A ed	(184)
§ A.1 引论	(185)

§ A.2	演示	(186)
§ A.3	基本概念	(187)
§ A.4	命令	(187)
§ A.5	上下文与正规表达式.....	(202)
§ A.6	加快编辑速度	(211)
§ A.7	用编辑程序进行分解和粘连	(214)
§ A.8	编辑命令程序	(215)
§ A.9	命令一览	(216)

- § 2.3.23 退出编辑程序
- § 2.3.24 系列文件的编辑
- § 2.3.25 不必退出编辑程序,编辑一新文件
- § 2.3.26 暂时退出编辑程序;shell 转换命令
- § 2.3.27 系列行定位命令的完成
- § 2.3.28 显示用户正在编辑的文件名
- § 2.3.29 显示用户正在编辑的行号和行内容

§ 2.4 解决一些常见的问题

§ 2.5 建立用户环境

- § 2.5.1 设置终端类型
- § 2.5.2 设置任选项;set 命令
- § 2.5.3 显示制表符和行末字符:list
- § 2.5.4 检索命令中不分大小写;ignorecase
- § 2.5.5 显示行标号命令:number
- § 2.5.6 显示变换行数命令;report
- § 2.5.7 改变终端类型命令;term
- § 2.5.8 缩短错误信息命令;terse
- § 2.5.9 关闭警告命令;warn
- § 2.5.10 容许检索中的特殊字符命令;nomagic
- § 2.5.11 检索约束命令;wrapscan
- § 2.5.12 打开信息通道命令;mesg
- § 2.5.13 规格化用户环境;.exrc 文件

§ 2.6 指令汇总

§ 2.1 绪言

所有 ASCII 文文件,如一程序或一文档,均可以应用屏幕编辑来建立和修改。适用于 XENIX 系统的屏幕编辑程序有两种:它们是 ed 和 vi。ed 在这本手册的附录 A 中将讨论到。

vi(取自 visual,意为直观)将行定位和屏幕定位两种方式结合成为有效的正文编辑操作整体,它将满足任何正文编辑的需要。

本章第一部分是示范举例,这将提供给读者一些有关的直接经验,该部分介绍了一些基本概念,在真正学会使用 vi 之前,必须熟悉这些概念,该部分还展示给读者怎样完成简单的编辑功能。第二部分是参考部分,告诉用户如何完成具体的编辑任务。第三部分描述了如何建立起用户的 vi 环境和如何设置选择项。第四部分是命令的汇总。

由于 vi 是如此有效的编辑程序,因而它具有比用户一次能学全的多得多的命令。如果你以前没使用过正文编辑程序,最好的掌握途径是先把在示范举例中讲的概念和操作完全弄清楚,然后再针对你要完成的特定任务参考第二部分。完成一个给定任务的所有步骤在每一部分都有解释,因此部分知识在文章中重复数次。用户对基本 vi 操作熟悉之后,就能够容易地懂得如何学会更高级更深入的功能。

如果用户以前使用过正文编辑器,用户可以直接查阅本章的特定任务部分。由学习 vi 的功能开始,以后用户将经常应用它。如果你是经验丰富的 vi 使用者,你可以不学这一章,而使用在《XENIX 参考手册》(Xenix Reference Manual)中讲的 vi(c)编辑器。

§ 2.2 演示范例

以下的演示实验将使用户获得使用 vi 的感性认识,并且介绍了一些基础概念。这些概念是在用户学会更复杂的操作之前必须弄懂的。用户能够学会如何进入和退出编辑程序,插入和删除正文,检索和模式替代,及如何将其它文件插入正文。此演示实验需要一个小时,需要记住的是,学习 vi 的最好方法是实际使用 vi,因此不要惧怕实验。

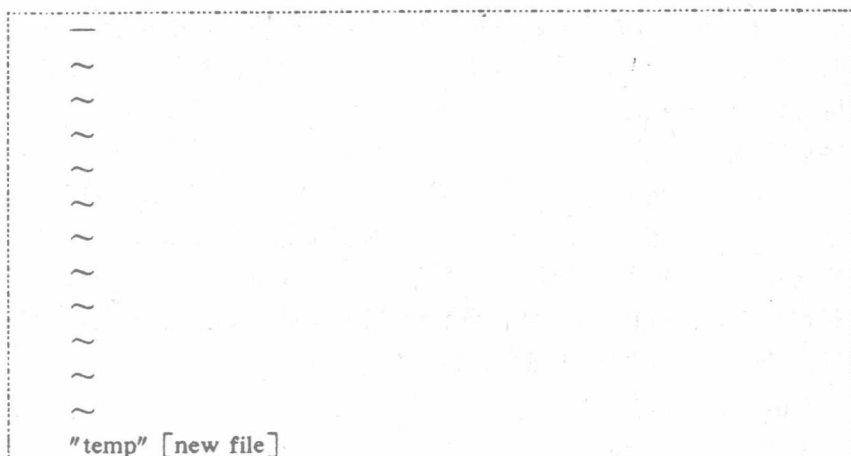
在开始演示实验前,须确信用户的终端已适当地设置起来,参看 2.5.1 节,“设置终端类型”,因为有关设置终端的许多知识是针对使用 vi 的。

§ 2.2.1 进入编辑程序

进入编辑程序并建立一个名字为 temp 的文件,须键入:

```
vi temp
```

屏幕显示如下:



注意：我们用了十二行显示屏幕以节省空间，实际上，vi 可利用用户所有的任意大小的屏幕。

用户开始编辑一文件的副本。直到存贮起来为止，文件本身不会被修改，文件存贮在演示实验后边部分讲到。显示的第一行也是文件的唯一行，由光标标出。如上所述光标显示一强调字符，本章中，光标显示在某字符上，则该字符用方括号 [] 括起。

光标所在行称之为当前行。

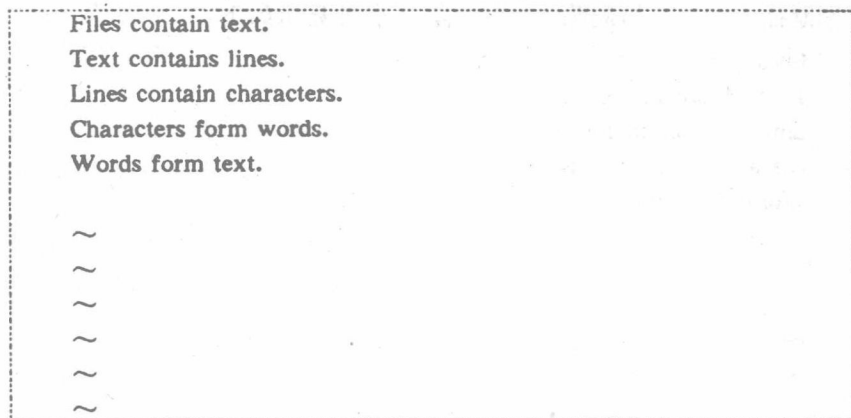
带有字符“~”的行不是文件的一部分：这些行只在屏幕上显示，而非文件中的行。

§ 2.2.2 插入正文

开始，先应用插入(i)命令，在文件 temp 中建立部分正文，因此，键入：

i

然后，键入下列五行形成正文，以备实验用到，键入每行之后，按 RETURN 键开始另一行。如果键入出错，使用 BKSP 键删去错误，重新敲入正确的词。



当你完成之后按 ESCAPE 键(ESC 键为其缩写)。象多数 vi 命令，i 命令不在屏幕上显示，命令本身使用户从控制方式转换到插入方式。当用户在插入方式时，键入的每个字符都显示在屏幕上。在控制方式时，键入的字符不被置在文件中作为正文，它们被解释为在文件上执行的命令。如果用户不能肯定工作在哪种方式，那么按 ESC 键直到听到铃声。听到铃声时，用户就工作在控制方式了。

一旦在插入方式,键入的字符都被插入到文件中,这些字符不被解释为 vi 命令,退出插入方式重新进入控制方式时,总是按 ESC 键。两种方式之间的转换在 vi 中经常用到,现在就习惯使用它是非常重要的。

§ 2.2.3 命令的重复

接下来在 vi 中经常使用的一条命令是:重复命令。重复命令是重复最后一次做的插入或删除命令。前面我们刚执行了插入命令,重复命令重复所做的插入,复制插入的正文,通过键入句号或点(.),重复命令得以执行,因此,正文要增加同样五行,可键入". "。重复命令的执行是相对于光标所在行,并且在当前行下面开始插入正文(记住,当前行一直是指光标所在行)。键入点(.)之后,屏幕显示内容如下示:

```
Files contain text.
Text contains lines.
Lines contain characters.
Characters form words.
Words form text.
Files contain text.
Text contains lines.
Lines contain characters.
Characters form words.
Words form text.
—
~
```

§ 2.2.4 取消命令

另外一个非常有用(而且开始时经常需要经常用到的)指令是取消(u)命令,按键

u

可以注意到刚刚插入的五行被删除或“被取消”。屏幕显示如下:

```
Files contain text.
Text contains lines.
Lines contain characters.
characters form words.
Words form text.
—
~
~
~
~
~
~
```

现在,重新键入:

u

被删去的五行又被重新插入。这条取消命令可以有效地从错误的删除或插入中恢复到正确状态。

§ 2.2.5 移动光标

现在我们来学习如何在屏幕上移动光标。除箭头键以外,以下字母键也控制光标移动:

h 左移

l 右移

k 上移

j 下移

字母键的选择是根据它们在键盘上的相对位置。记住:光标移动键只工作在控制方式。

键入 L 指令使光标移动到屏幕的最后一行(在本例中值得注意的是:L 命令使光标移到屏幕的最后一行,而 l 命令使光标右移一字符)。接下来,用 goto 命令 G 移动光标到文件的最后一行。如果键入命令 2G,则光标移动到文件的第二行;如果文件有 10,000 行,键入命令 8888G 之后,光标移动到第 8888 行(如果文件有 600 行,则键入命令 800G 时,光标并不移动)。

上述的光标移动命令使你可以在演示实验中很方便地移动光标。其它的可用的光标移动指令是:

Ctrl-u 上滚 1/2 屏

Ctrl-d 下滚 1/2 屏

Ctrl-f 向前滚一整屏

ctrl-b 向后滚一整屏

§ 2.2.6 删除

前面我们已讲述如何插入和建立正文,如何在文件内移动光标,这里我们讲述如何删除正文。正如以下所述,许多删除命令是与光标移动命令结合在一起的,最常用的删除命令如下:

dd 删除当前行(光标所在行),不管光标所在具体位置。

dw 删除光标所在词。如果光标在词的中部,删除该词自光标以后的字母。

x 删除光标所在字母。

d\$ 删除自光标到本行末的正文。

D 删除自光标到本行末的正文。

d0 删除本行开始到光标处的正文。

· 重复最后一个变换指令(只有最后指令是删除指令时才能用此命令)。

要学会如何使用这些删除命令,我们将删除演示文件的各个不同部分。开始,按 ESC 键以确保工作在命令方式,尔后键入命令 1G 移动到文件的第一行。

开始,文件显示如下所示:

```
[F]iles contain text.  
Text contains lines.  
Lines contain characters.  
Characters form worrds.  
Words form text.  
Words form text.  
Text contains lines.  
Lines contain characters.  
Chatacters form words.  
Words form text.  
~  
~
```

要删去第一行,键入命令:

dd

文件显示将如下所示:

```
[T]ext contains lines.  
Lines contain characters.  
Characters form words.  
Words form text.  
Files contain text.  
Text contains lines.  
Lines contain charaters.  
Characters form words.  
Words form text.  
~  
~  
~
```

删除光标所在词 Text,键入命令:

dw

删除之后,文件显示如下:

```
[c]ontains lines.  
Lines contain characters.  
Characters form words.  
Words form text.  
Files contain text.  
Text contains lines.  
Lines contain characters.  
Character form words.  
Words form text.  
~  
~  
~
```

要删除光标所在的字符,可键入命令:

x

文件显示如下:

```

[O]toins lines.
Lines contain characters.
Characters form words.
Words form text.
Files contain text.
Text contains lines.
Lines contain characters.
Characters form words.
Words form text.

```

现在,键入 w 命令使光标移动到第一行的词 lines 的开始,然后,删除当前行光标以后所有内容,键入:

d\$

文件显示如下所示:

```

ontains_
Text contains lines.
Lines contain characters.
Characters form words.
Words form text.
Files contain text.
Text contains lines.
Lines contain characters.
Words form text.

```

要删除该行光标前的所有字符,键入命令: .d0

屏幕显示如下所示,该行只留下一空格:

```

_
Lines contain chatacters.
Files contain text,
Text contains lines.
Chatacters form words.
Words form text.
Lines contain characters.
Characters form words.
Words form text.

```

为了复习前面所讲的插入命令,我们恢复文件的前两行。

按 i 键进入插入方式,键入以下两句话:

Files contain text.

Text contains lines.

按 ESC 键返回到命令方式。

§ 2.2.7 模式检索

vi 提供了检索字符表达式的功能。具体做法是在斜线(/)后面写上你要检索的表达式,以按 RETURN 键表示该表达式的结束。

例如,确信用户处于命令方式(按 ESC 键)状态,然后键入命令:

H

即可移动光标到屏幕的上端,键入命令:

/char

注意先不要按 RETURN 键,屏幕显示如下所示:

```
Files contain text.  
Text contains lines.  
Lines contain characters.  
Charaters form words.  
Words form text.  
Files contain text.  
Text contains lines.  
Lines contain characters.  
Characters form words.  
Wyrds form text. ~  
/char_
```

按 RETURN 键,光标移动到第三行单词 characters 的第一个字符 c 上。如继续检索寻找 char 模式,则按 n 键(取自 next 字头)。光标将移动到第八行单词 characters 的第一个字符 c 上。如果继续按 n 键,vi 检索超过文件末尾,卷回到文件开始,重新找到第三行的 char。

注意 斜线符(/)和要检索的表达式显示在屏幕的底部,屏幕上的最后一行是 vi 的状态显示行(简称状态行)。

状态行出现在屏幕的最后一行。状态行用来显示某些信息,包括要检索的表达式,行定位命令(此命令在后面解释)及错误信息。

例如,要得到关于文件的状态信息,按 Ctrl-g 键,屏幕显示如下:

```
Files contain text.
Text contains lines.
Lines contain charaters.
Characters form words.
Words form text.
Files contain text.
Text contains lines.
Lines contain [c]haracters.
Characters form words.
Words form text.
```

~

"temp" [Modified] line 4 of 10 --40%--

底部的状态行给出用户正在编辑的文件名,文件是否已经被改过,当前行行号及文件的总行数,并且用百分数表示用户在文件中的当前行位置。

§ 2.2.8 检索和替代

假如要把演示文件中所有出现“text”的地方都变换为“documents”。我们可使用一条命令完成,而不用先去检索“text”,再删掉“text”,再插入“documents”。迄今为止,前面所学过的命令都是屏幕定位的。能够起到不止一种作用(检索和替代)的命令是行定位(line-oriented)命令。

屏幕定位(Screen-oriented)命令从光标所在处开始执行。用户不需要告诉计算机从何位置开始执行命令,命令从光标所在处开始执行。行定位(line-oriented)命令需要用户给出一确定位置(称为“地址”),命令在此地址处执行。

屏幕定位命令很容易实现,而且马上提供反馈信息:改变的内容显示在屏幕上。

行定位命令实现起来比较复杂,但可独立于光标执行,而且在同一时间可在同一文件中多处执行。

所有的行定位命令之前都有一冒号(:)作为在状态行上的提示符,行定位命令本身从该行输入以“回车”(RETURN)终止。

在本章,所有行定位命令都包括冒号(:)作为命令的一部分

:1,\$s/text/documents/g

此命令意味着:“从第一行(1)到文件末尾(\$),找出 text 并用 documents 代替它(s/text/documents),对每一行都进行同样的操作(g)”。

按“回车”键(RETURN),屏幕显示如下:

```
Files contain documents.
Text contains lines.
Lines contain characters.
Characters form words.
Words form documents.
Files contain documents.
Text contains lines.
Lines contain characters.
Characters form words.
[W]ords form documents.
~
~
```

注意,第二行和第八行的 Text 没被替换,这种情况在检索中需注意。

为了练习,用 Undo 命令变 documents 恢复到 text 形式,键入命令:

u

屏幕显示如下:

```
[F]iles contain text.
Text contains lines.
Lines contain characters.
Characters form words.
Words form text.
Text contains lines.
Lines contain characters.
Characters form words.
Words form text.
~
```

§ 2.2.9 退出 vi

用户所做的所有编辑工作已经改变了文件的副本,不再是当初进入 vi 时所定义的文件名为 temp 的文件了,为了存储改变的文件,退出编辑程序,返回到 XENIX 的 shell,键入命令:

:x

记住按“回车(RETURN 键)”,文件名、总行数和文件所包含的字符数都显示在状态行上:

```
"temp" [New file] 10 lines, 214 characters.
```

然后 XENIX 提示符出现。

§ 2.2.10 将其他文件加入正文

在这一部分我们将建立一个新文件。而且可以从另一个文件插入正文到新文件。开始,通过键入以下指令建立一新文件:

```
vi practice
```

此文件是空的。下面从 temp 文件复制到正文,用行定位 Read 命令置入 practice 文件中。按 ESC 键确信工作在命令方式,然后键入: