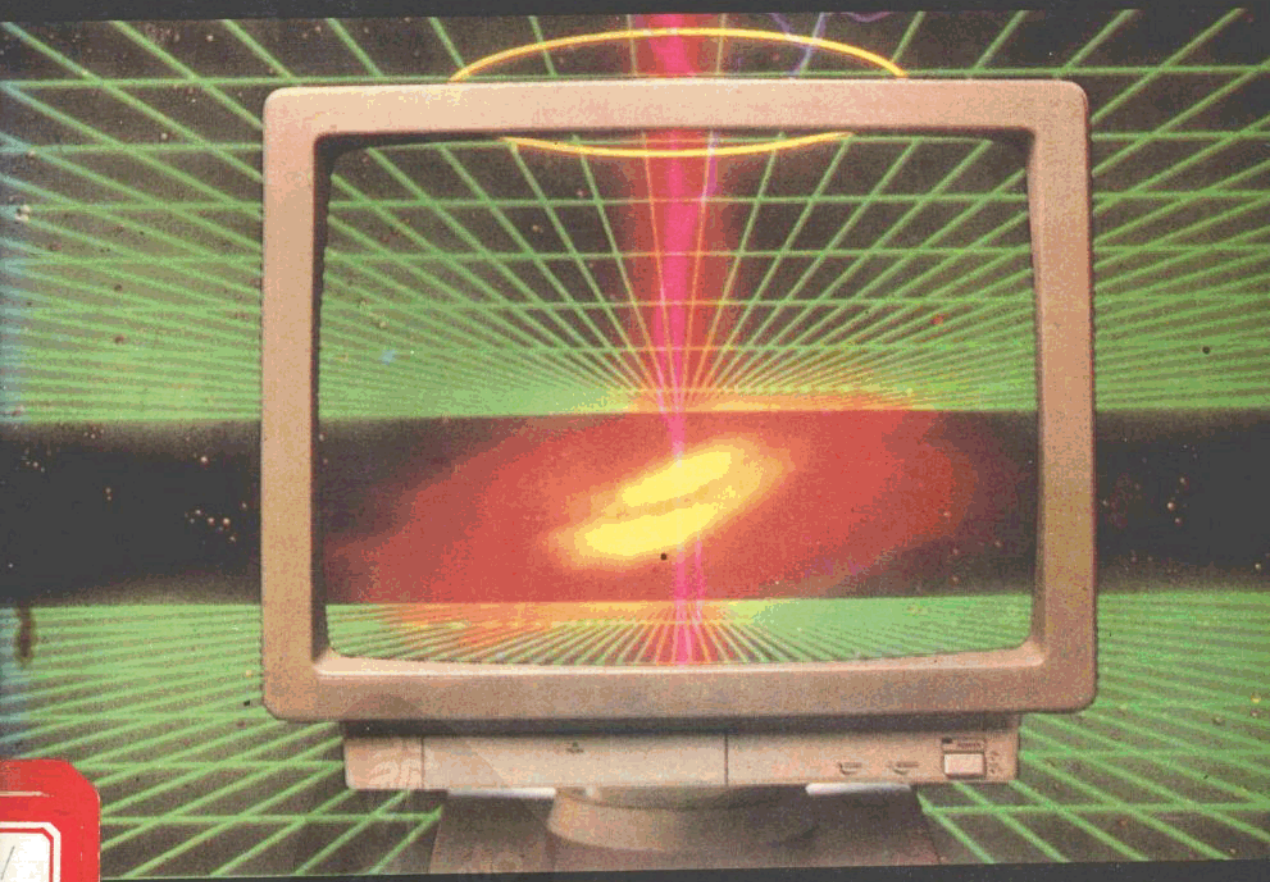
适用于IBM PC AT 286 386及其兼容机

SCOXENIX 系统V

技术丛书

（共五册）

用户指南



上海电子计算机厂
北京希望电脑公司

2

第一章 绪 言

§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缉程序 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 系统之间用拔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。

此为试读，需要完整PDF请访问：www.ertongbook.com

§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：

```
~
~
~
~
~
~
~
~
~
~
~
~
"temp" [new file]
```

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。

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

~
~
~
~
~
~
```

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：

```
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：

此为试读，需要完整PDF请访问：www.ertongbook.com

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,确信用户外于命令方式(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.
Wrds 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目，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，而不用去先检索“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:

:r temp

文件显示如下：

```
[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.
~
```

temp 的正文已被复制并插入到当前文件 practice 中。文件开头有一空行。移动光标到空行，用 dd 命令删除空行。

§2.2.11 暂时退出 vi

vi 允许用户在正编辑着的文件外执行命令，例如 date 命令。要知道日期和时间，键入命令：

```
:! date
```

按 RETURN 键，将显示日期。系统提示用户按 RETURN 键重新进入命令方式。用户不妨试一下，屏幕显示如下。

```
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.
~
:! date
Mon Jan 9 16:33:37 pst 1985
[Hit return to continue]—
```

§2.2.12 改变显示形式

除了上面描述的一系列编辑命令，当用户调用 vi 或进行编辑时，还可以设置若干任选项，这些任选项允许用户控制编辑参数，如对行号显示、检索时是否有特殊情况(如字母大小写)需注意等等的控制，在本小节，将介绍如何启用自动编排行号功能以及如何列出包括所有任选项的表。

要启用自动编排行标号功能，键入命令：

```
:set number
```

按“回车(RETURN)”键。屏幕重新显示，行号出现在正文的左侧。具体显示如下所示：

```
1 Files contain text.
2 text contains lines.
3 Lines contain characters.
4 Characters form words.
5 Words form text.
6 Files contains lines.
7 Text contains lines.
8 Lines contain characters.
9 Characters form words.
10 Words form text.
~~
```

用户要得到所有可用任选项完整的表，可键入命令：

```
:set all
```

然后按“回车(RETURN)”键即可。如何设置这些任选项将在 2.5 节“建立用户环境”中讲到，但用户意识到其存在是非常重要的。根据用户工作的机器和用户的选择，用户可以改变许多选择项的缺省设置。

§2.2.13 删除编辑过程

最后要退出 Vi 而不必存贮 practice 文件，则键入命令：

```
:q!
```

然后按 RETURN 键即可。这就删除了用户对文件 practice 所做的所有修改，由于 proctice 是个新文件，因而被删除。接着提示语句出现，如果在编辑之前文件 practice 已存在，则用户修改的内容不被承认，但原文件仍将存在而不是被删除。

到此为止，演示部分已结束，前面已讲述过了如何进入和退出 vi，如何插入和删除正文、移动光标、检索和替代、如何执行行定位命令、如何从其他文件复制正文和如何删除编辑中所修改过的内容。

当然还有许多命令需要掌握，但使用 vi 的基本命令均已讲述过了。以下几部分将提供给读者关于这些命令的详细介绍和其他的 vi 命令及要点。

§2.3 编辑任务

下面讲述的 29 小节解释了如何完成通常的编辑作业。根据每一部分的讲授，读者应该能够完成指定的作业编辑工作。完成几个任务所需要的功能介绍在每次用到时都讲述，因而一些信息是重复提到的。

§2.3.1 如何进入编辑程序

有几种方式可以开始编辑，这取决于用户自己的操作。这一部分讲述如何用一个文件名开始或进入编辑程序。vi 编辑程序可同时对一系列文件进行操作，可参看 2.3.24 节“系列文件的编辑”(“Editing a Series of Files”)。

▲用文件名进入 vi

进入 Vi 的最普通的方法是键入命令 vi 和要编辑的文件名：

```
vi filename
```

如果上述文件名前面并没存在，则一个新的空文件就建立了。

▲由特定行进入 vi

用户也可以由文件中的特定位置进入编辑文件。例如，希望从文件的第 100 行开始编辑，则键入命令：

vi +100 filename.

即可。光标则被调至编辑文件的第 100 行。

▲由特定词进入 vi

如果希望从一特定词最初出现的位置开始编辑，则键入命令：

vi +/word filenamc.

即可。光标则被调至文件中该词首次出现的位置。例如，希望对文件 temp 从特定词 contain 首次出现的位置进行编辑，则键入命令：

vi +/contain temp

即可。

§2.3.2 光标的移动

光标移动键允许用户在文件中随意移动光标。光标移动只能在命令方式下才有效。

△用下列字符移动光标：h，j，k，l，Space，Bksp

空格键和 l 键使光标向后移动一指定的字符数。BKSP 键和 h 键使光标向后移动一指定的字符数。如果无指定字符数，则光标移动一个字符。例如：使光标后移 4 个字符，则键入命令：

4h

用户还可以使光光标移动到当前行中一指定的字符上。F 向后移动光标到指定字符上，f 则向前。光标停留在指定字符上。例如，将光标向后移动到当前行中最接近的 p 字符上，则键入命令：

Fp

向前移动到最接近的 p 字符上，则键入：

fp

T 和 t 键的功能与 f 和 F 键类似，只不过 T 和 t 键立即把光标调至指定字符的前面，例如，要把光标调回到当前行中最接近的 p 字符的下一位置，则键入：

Tp

如果 p 是在单词 telephone 中，则光标被调至字符 h 上。

当使用上述这些命令时，光标始终保持在同一行上，如果指定一字符数比当前行中的字符数大，光标也不会移动到前一行或后一行。

△光标在单词间的移动命令：w，W，b，B，e，E

w 键使光标向前移动指定数目的单词的开头。标点和非字母字符(如！@ # $ % ^ & * ( ) — + { } [ ] \ ' > /)被认为是单词，因此如果一词后面有一逗号(,)，则逗号也计算在指定单词数之中。

例如，现在光标在此句的第一个字母上：

No，I didn't know he had returned.

如果键入命令：

6w

此为试读，需要完整PDF请访问：www.ertongbook.com