

IBM 操作系统

(中二)



VM/370

(VM/SP)

试验所

中国科学院成都计算机应用研究所服务公司

目 录

第七章 XEDIT使用方法 (1)

第一节 XEDIT全屏幕文本处理 (1)

- 1. 编辑文件 (1)
 - 1.1 XEDIT命令 (1)
 - 1.2 屏幕布局 (1)
- 2. 输入数据 (3)
 - 2.1 INPUT子命令 (3)
 - 2.2 POWER子命令 (4)
- 3. 程序功能键的使用 (5)
 - 3.1 行的剖分和连接 (8)
 - 3.2 屏幕的前后翻转 (8)
 - 3.3 子命令的再显示 (8)
 - 3.4 子命令的再执行 (9)
 - 3.5 在一行中插入字 (9)
- 4. 前缀子命令的使用 (9)
 - 4.1 行的添加与删除 (9)
 - 4.2 行的重复 (10)
 - 4.3 行的移动和拷贝 (10)
 - 4.4 当前行的设置 (11)
 - 4.5 前缀子命令的清除 (11)
- 5. 在屏幕上移动文件 (11)
 - 5.1 BACKWARD与FORWARD (11)
 - 5.2 TOP与BOTTOM (12)
 - 5.3 DOWN与UP (12)
- 6. 文件修改 (12)
 - 6.1 CLOCATE子命令 (12)
 - 6.2 CHANGE子命令 (13)
 - 6.3 选择修改 (13)
 - 6.4 全局修改 (14)
 - 6.5 CINSERT子命令 (14)
 - 6.6 CFIRST子命令 (16)

7. TAB设置	(16)
8. 编辑会话结束	(16)
8.1 FILE子命令	(16)
8.2 QUIT子命令	(17)
8.3 SET AUTOSAVE	(17)
9. 从别的文件插数据	(17)
9.1 插入整个文件	(18)
9.2 插入另一文件的一部分	(18)
10. 请求帮助	(19)
11. 常用XEDIT子命令	(19)
12. 示例	(20)
12.1 建立文件	(20)
12.2 使用Power Typing	(21)
12.3 使用前缀子命令	(22)
12.4 修改文件	(23)
12.5 提取和连接文件	(24)
第二节 目标使用	(25)
1. 什么是目标	(25)
2. 为把某一行变为当前行而使用目标	(26)
2.1 独自打入的目标	(26)
2.2 作LOCATE子命令操作数的目标	(27)
2.3 置于子命令前的目标	(27)
3. 把目标作为子命令操作数	(28)
4. 目标的类型	(28)
4.1 作为绝对行号的目标	(28)
4.2 作为当前行相对位移量的目标	(28)
4.3 作为行名的目标	(29)
4.4 作为简单串表达式的目标	(30)
4.5 作为复合串表达式的目标	(31)
5. 列目标使用方法	(33)
第三节 编辑多个文件	(35)
1. XEDIT子命令	(35)
2. 建立存贮中的文件环	(35)
3. 编辑环中的文件	(35)
4. 结束编辑会话	(36)
5. 多个逻辑屏幕	(36)
5.1 SET SCREEN子命令	(37)
5.2 同一文件的多种视图	(37)

5.3 不同文件的多个视图	(37)
第四节 屏幕调整	(38)
第五节 宏语言	(39)
1. XEDIT宏命令及建立	(39)
2. 宏命令中的XEDIT子命令	(39)
2.1 Editor与EXEC2 通讯	(40)
2.2 编辑屏幕的数据显示	(40)
2.3 编辑变量的保留与恢复	(41)
2.4 如何发CMS和CP命令	(41)
2.5 避免名字的混淆	(41)
3. XEDIT宏命令示例	(42)
4. 编辑用的PROFILE宏命令	(44)
4.1 PROFILE宏命令的执行	(45)
4.2 编写PROFILE宏命令	(45)
4.3 PROFILE宏命令实例	(45)
5. 前缀宏命令	(46)
5.1 宏命令的信息传送	(47)
5.2 源串的检查	(47)
5.3 传送信息的使用	(48)
5.4 成组行的处理	(48)
5.5 变元串的检查	(49)
5.6 光标定位	(50)
5.7 XEDIT子命令的使用	(50)
5.8 前缀宏命令L的完整清单	(50)

第八章 XEDIT 命令及宏命令

第一节 XEDIT子命令	(51)
ADD	(51)
ALL	(52)
ALTER	(52)
BACKWARD	(53)
BOTTOM	(53)
CANCEL	(53)
CAPPEND	(54)
CDELETE	(54)
CFIRST	(54)

CHANGE	(55)
CINSERT	(56)
CLAST	(56)
CLOCATE	(56)
CMS.....	(57)
CMSG	(57)
COMMAND	(58)
COMPRESS	(58)
COPY	(58)
COUNT	(59)
COVERLAY.....	(59)
CP	(59)
CREPLACE	(60)
CURSOR	(60)
DELETE	(61)
DOWN.....	(61)
DUPLICAT	(61)
EMSG	(62)
EXPAND	(62)
EXTRACT.....	(63)
FILE	(68)
FIND	(69)
FINDUP... ..	(69)
FORWARD	(69)
GET.....	(70)
HELP	(70)
HEXTYPE.....	(70)
INPUT	(71)
JUIN	(71)
LEFT	(72)
LOAD.....	(72)
LOCATE	(73)
LOWERAS	(74)
LPREFIX	(74)
MERGE	(75)
MODIFY	(76)
MOVE	(77)
MSG	(77)
NEXT	(77)
NFIND	(78)

NFINDUP	(78)
OVERLAY	(78)
PARSE	(79)
POWERINP.....	(79)
PRESERVE.....	(80)
PURGE.....	(80)
PUT	(81)
PUTD	(81)
QUERY.....	(81)
QUIT.....	(85)
RECOVER	(86)
REFRESH	(86)
RESTORE	(86)
RGTLEFT	(86)
RENUM	(87)
REPEAT	(87)
REPLACE	(87)
RESET.....	(87)
RIGHT.....	(88)
SAVE	(88)
SET	(88)
SET ALT	(88)
SET APL	(89)
SET ARBCHAR	(89)
SET AUTOSAVE	(90)
SET CASE.....	(90)
SET COLOR	(91)
SET CMDLINE	(92)
SET COLPTR	(92)
SET CTLCHAR	(92)
SET CURLINE.....	(93)
SET DISPLAY.....	(93)
SET ENTER	(94)
SET ESCAPE	(94)
SET FILLER.....	(95)
SET FNAME.....	(95)
SET FTYPE	(95)
SET FMODE.....	(95)
SET FULLREAD	(96)
SET HEX	(96)

SET IMAGE	(96)
SET IMPCMSCP	(97)
SET LASTLORC	(97)
SET MACRO	(98)
SET MASK	(98)
SET MAGLINE	(98)
SET NONDISP	(99)
SET NULLS	(99)
SET NUMBER	(100)
SET PACK	(100)
SET PA _n	(100)
SET PENDING	(101)
SET PF _n	(103)
SET POINT	(103)
SET PREFIX	(104)
SET RANGE	(104)
SET RECFM	(104)
SET REMOTE	(105)
SET RESERVED	(105)
SET SCALE	(106)
SET SCOPE	(106)
SET SCREEN	(107)
SET SELECT	(109)
SET SERIAL	(110)
SET SHADOW	(110)
SET SIDCODE	(111)
SET SPAN	(111)
SET SPILL	(112)
SET STAY	(113)
SET STREAM	(113)
SET SYNONYM	(113)
SET TABLINE	(114)
SET TABS	(115)
SET TERMINAL	(115)
SET TEXT	(115)
SET TOFEQF	(116)
SET TRANSLAT	(116)
SET TRUNC	(117)
SET VARBLANK	(117)
SET VERIFY	(117)

SET WRAP	(118)
SET ZONE	(118)
SET =	(119)
SHIFT	(119)
SORT	(119)
SOS	(120)
SPLIT	(120)
SPLTJOIN	(121)
STACK	(121)
STATUS	(122)
TOP	(122)
TYPE	(122)
UP	(123)
UPPERCAS	(123)
^EDIT	(123)
&	(124)
=	(124)
?	(124)

第二节 前缀子命令 (124)

A	(124)
C	(125)
D	(125)
E	(125)
F	(126)
I	(126)
M	(126)
P	(126)
//	(126)
/	(127)
SCALE	(127)
TABL	(127)
,XXXX	(127)
S	(127)
< (左移位)	(128)
> (右移位)	(128)

第三节 补充材料 (129)

1. 文件类的预置值	(129)
1.1 SET子命令各自选项预置值	(129)

1.2 Tab制表列的预置值	(130)
2. 选择行编辑子命令的作用	(130)
2.1 自动选择级的分配	(131)
2.2 SCOPE对子命令的影响	(131)
2.3 从屏蔽行发前缀子命令和宏命令	(132)
3. 优化宏命令	(133)

第九章 FLIST与BROWSE

第一节 FLIST功能	(144)
1. FLIST命令	(134)
2. FLIST子命令	(135)
3. 环境转换键	(137)
4. 程序功能键	(137)
5. 用户输入区使用方法	(137)
6. FLIST的PROFILE文件	(138)
7. FLIST返回码	(139)
第二节 BROWSE功能	(139)
1. BROWSE命令	(139)
2. BROWSE子命令	(140)
3. 环境转换与程序功能键	(142)
4. BROWSE PROFILE的使用	(142)
5. BROWSE返回码	(143)

第七章 XEDIT使用方法

XEDIT是随VM/SP带的编辑程序产品，它为文本处理和程序的开发提供了广泛的编辑功能，比VM/370的编辑程序更方便、更灵活。XEDIT既有全屏幕方式又有行编辑方式，可以用于显示终端也可用于打字机型终端。由于目前极少使用打字机型终端，故在本章中不单独介绍打字终端。打字终端上使用XEDIT时，除不能显示一整屏外，其它与显示终端相似。

本章按实际使用介绍XEDIT，第八章详细介绍XEDIT的各个子命令。

第一节 XEDIT全屏幕文本处理

本节介绍XEDIT的文本处理，看完本节后，你对编辑程序就会有一定的了解。这里出现的子命令包括XEDIT子命令的一个子集，用这些子命令你可以建立文件，打入数据，管理屏幕，改变文件以及在文件之间传递数据。

编辑程序还有许多其它功能，它们在本书的其余各节及下一章叙述。

1. 编辑文件

对文件进行编辑意味着对磁盘上的CMS文件进行改变，添加或删除处理。这些变动都是交互式进行的：用户指示编辑程序进行一次修改，编辑程序照此执行，然后用户再要求另一次修改。可以编辑还不存在的文件，那就是建立新文件。

1.1 XEDIT命令

在Logon上VM/SP和进入CMS环境后，也就准备好进入编辑环境并开始建立文件。编辑程序由CMS命令XEDIT调用，格式为：

```
XEDIT filename filetype filemode
```

在图7—1中，编辑程序由下面的命令调用：

```
XEDIT INVENTOR SCRIPT A
```

在看如何往文件中打入数据之前，先看看图7—1所示的屏幕布局。（不同版本的VM/SP稍有差异。）

1.2 屏幕布局

①文件标识行

屏幕的第一行标志出正在编辑的文件，该行显示出下列信息：

- 文件名，文件类、文件型，如果未规定文件型，编辑程序认为文件型是“A1”，这就表示该文件是你的A盘上文件。
- 记录格式和记录长度，本例中的记录格式和记录长度（V132）表示在这个文件的记录为变长记录，行的长度限制为132个字符。因此，文件行可以比屏幕的一行长。

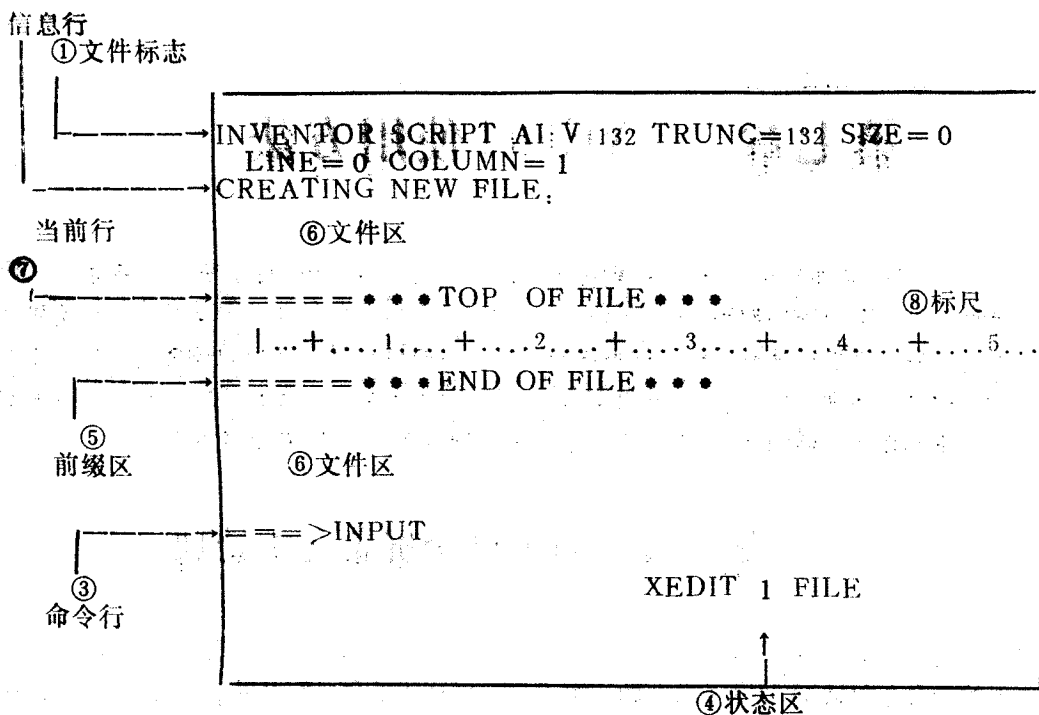


图 7—1 屏幕布局

- c、截断列 (TRUNC=)，注意截断列与记录长度 (132) 相同。因为文件行只能有 132 个字符长，所以在 132 个字符以外打入的任何数据都被截去。
- d、文件中的现有行数 (SIZE=) (由于我们还没有在文件中打入数据，故行数为 0)
- e、文件的当前行号 (LINE=) (见下面的⑦)
- f、列指针位置 (COLUMN)= (见下面的⑧)

②信息行

编辑程序通过在屏幕的第二行上显示信息与用户进行通讯。在你出错时会通知你出错信息，它们也可以提供其它信息，在图 7—1 中，信息行表明用户正在建立新文件。

③命令行

屏幕底部的大箭头 (==>) 指到命令输入区。用户与编辑程序通讯的方法之一就是在这一行打入 XEDIT 子命令。子命令可以用大写或小写形式或者混合形式打入，有许多子命令还有缩写形式。例如：“INPUT” “Input” 和 “i” 都是打入 INPUT 子命令的有效形式。

在命令行中打入子命令之后，必须按 ENTER 键去执行子命令。图 7—1 说明在命令行中打入 “INPUT”。(为了把光标从屏幕上的任何地方移到此命令行，只要按 ENTER 键即可。)

④状态区

右下角显示编辑会话的当前状态，例如，编辑方式或输入方式。这个区也显示正在编辑的文件个数。图 7—1 的状态区表示有一个文件正在编辑。

⑤前缀区

前缀区是屏幕的左边 5 列，显示 5 个等号 (=====)。文件的各行都有与它相应的前缀区。在任何一行的前缀区可以打入一个字符的命令 (称为前缀子命令) 去执行各种编辑

任务, 比如用D去删除一行。

⑥文件区

屏幕的其余部分用于显示文件。用户可以通过把光标移到任何一行下面并打入新的字符或使用特殊键插入或删除字符的方法来改变文件。在按ENTER键之前, 可以在显示的行上按用户的要求作很多改动。按ENTER键时, 系统对原文件在虚存中的拷贝进行相应的改动, 在编辑会话结束时, FILE子命令把这些改动永久记录在该文件留在磁盘上的拷贝中。

文件很大, 一屏装不完时, 可以利用几种子命令翻转屏幕使文件能向前或向后移动。

⑦当前行

当前行是在屏幕中间(标尺上面)的文件行, 它的亮度比其它行高。在图7-1中, 当前行是“TOP OF FILE”行; 此时, 文件还未包含数据。当前行是一个重要的概念, 因为大多数子命令都从当前行开始执行他们的功能。在上下翻转屏幕时, 当前行也要随着改变。当前行改变时, 我们就说行指针(在屏幕上看上见)移动了。很多XEDIT子命令从当前行开始执行其功能, 做完后移动行指针。

⑧标尺(Scale)

标尺出现在当前行下面帮助用户编辑, 就像打字机上的边界刻度。图中屏幕第一列上的竖杠(|)是列指针。有些子命令在一行中从列指针开始执行它们的功能。可以用一些XEDIT子命令把列指针移到标尺的不同位置, 这些子命令将在后面讨论。列指针所指的列称为当前列。

2. 输入数据

在你打入XEDIT命令后就处于编辑方式。在编辑环境中, 能打入各种XEDIT子命令, 进一步利用input方式或Power typing方式可以把数据打入文件。

2.1 INPUT子命令

为了进入输入方式, 在命令行中打入下面的子命令:

```
===> INPUT
```

然后按ENTER键。此后你就可以在下半屏(标尺与命令行之间)的输入区域中打入数据。

图7-2显示的文件INVENTOR SCRIPT与图7-1相同。不过已打入了INPUT子命令还打入了几行数据到屏幕上。注意在输入方式下屏幕的变化: 前缀区(=====)消失; 信息行和状态区告诉你正处于输入方式; 命令行包含短语“INPUT ZONE”, 它标志输入区域的结尾并提醒你在输入方式中不能打入子命令。

在图7-2中, 整个输入区已满。为了保持输入方式并打入更多的数据, 按一次ENTER键。你所打入的行都移到屏幕的上半部, 最后打入的行成为新的当前行, 输入区域又可以打入更多的数据。

如果没有更多的行要打入, 再按一次ENTER键就退出输入方式而回到编辑方式。

在编辑会话期间的任何时候, 可以进入输入方式以便把新的数据行插入文件。正如你所见到的, 在打入INPUT子命令后, 编辑程序腾出空位让你在当前行后打入新的数据行。在这个例子中, 由于文件是新文件, INPUT是打入的第一个子命令, TOP OF FILE行是当前行。后面你将看到如何使任一行成为当前行让你在输入方式中把数据行插到文件的任何两个原有行之间。

```
INVENTOR SCRIPT AL V132 TRUNC=132 SIZE=12 LINE=0
COLUMN=1
INPUT MODE;
```

```
• • • TOP OF FILE • • •
```

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....
```

```
THE ELECTRONIC COMPUTER (1946)
```

```
• SP
```

```
THE WORLD'S FIRST ELECTRONIC COMPUTER WAS CALLED
ENIAC, ELECTRONIC NUMERICAL INTEGRATOR AND COMPUTER.
IT WAS BUILT BY A GROUP OF RESEARCHERS LED BY AMERICAN
PHYSICIST JOHN MAUCHLY AT THE UNIVERSITY OF
PENNSYLVANIA.
```

```
UNLIKE EARLIER COMPUTERS THIS ONE RAN ON RADIO TUBE S
18,000 OF THEM IN TOTAL.
```

```
IT FILLED A ROOM 30 FEET BY 50 FEET AND COST $ 400,000.
```

```
===> • • • INPUT ZONE • • •
```

```
INPUT—MODE 1 FILE
```

图 7—2 输入方式——打入数据

2.2 POWER子命令

打入一个大的文本，如象文章的一个长段，最容易的方法是使用“power typing”。为了使用power typing，打入子命令：

```
===> POWER
```

使用power typing的好处是可以随意打入数据就好像屏幕是一个很长的行。用户不必耽心行的长度或单词的长度——可以在屏幕的一行开始输入一个单词的字母而在下一行结束。事实上，如果用户是一个熟练的打字员，甚至可以不看屏幕。到了一行的末尾时，编辑程序自动卷绕到下一行的开始。用户可以连续打直到屏幕填满。

如果打满一屏后还想继续以power typing方式打入正文，按一次ENTER键，刚打的最后一行将显示在屏幕的顶部；其余为空白，让用户继续输入。

输入完毕时，按二次ENTER键就退出power typing方式回到编辑方式。编辑程序自动把数据分割为适当的屏幕行并重新安排被分开的单词。

在编辑会话期间的任何时刻可以通过打入POWER子命令的方法使用power typing。用power typing打入的数据插在当前行之后，这就像使用INPUT子命令一样。

2.2.1 数据的剖分

如果想要剖分以power typing方式打入的数据，即是要部分数据从一个新行开始（例如，一个新的段落为SCRIPT/V5控制字，它必须从第一列开始），可以在要从新行开始的数据前打入一个行结束字符，预置的行结束字符是（#）号。

例如，以power typing方式打入下列数据：

```
• SP# A pound sign causes the data to start on a new line. # SP
```

数据输入文件会成为：

```
===== • SP
```

=====
=====
===== A Pound Sign causes the data to start on a new line.

=====
===== . SP

2.2.2 字符的插入

如果要在一行中插入字符或空格而你又处于power typing方式,这时可以使用插入方式键。当字符插入之后,后面的数据都向右移动,就好像在一列火车中插入一个车箱。记住:在插完字符后按RESET键。

2.2.3 POWER TYPING的例子

图7-3(第一部分)描述同一文件,INVENTOR SCRIPT,但数据是在打入POWER子命令后以power typing方式打入的。在power typing方式中屏幕有一些变:前缀区和状态区消失;在打入POWER子命令时,当前行移到屏幕顶部,屏幕的其余部分可用于打入数据。注意看一个单词如何在一行开始而在下一行结束,整个屏幕可以用数据填满,但并不必须这样做。

从顶行数起的第8行上有二个#号,#号使后面的数据在进入文件时从新的一行开始。#号本身并不进入文件。

图7-3(第2部分)说明在按了二次ENTER键之后屏幕的内容。屏幕布局恢复编辑状态,单词和各行重新安排,#号后面的数据都从新行开始。

3. 程序功能键(PF)的使用

对各PF键设置了XEDIT子命令,当按PF键时执行相应功能。使用PF键节省了在命令行打子命令和按ENTER键的时间。

使用了命令

====> QUERY PF

可以显示PF键的设置情况。PF键的初始设置如下:

PF1	HELP MENU	PF7	BACKWARD
PF2	SOS LINEADD	PF8	FORWARD
PF3	QUIT	PF9	=
PF4	TABKEY	PF10	SPLIT CURSOR
PF5	SCHANGE 6	PF11	JOIN CURSOR
PF6	?	PF12	CURSOR COLUMN

这些是编辑程序预分配给各PF键的子命令。(由于profile宏文件的差异,PF键的初始设置可能有所不同。)如果想要改变一个或多个PF键的子命令,可以使用SET PF子命令,格式如下:

====> SET PFn subcommand

其中“n”是PF键的号数,subcommand是任一XEDIT子命令。例如:

====> SET PF1 INPUT

分配INPUT子命令给PF1键。按PF1键将立即使你处于输入方式。当你指定一个子命令给PF键时,这种设置只对当前编辑会话有效。在下一一次编辑会话中,仍恢复到上面显示的初始设置。

下面几段说明某些(初始设置的)PF键如何使用。另一些在适当的地方讨论。

INVENTOR SCRIPT A1 *** POWER TYPING ***

*** TOP OF FILE ***

THE WORLD'S FIRST ELECTRONIC COMPUTER WAS CALLED ENIAC, ELECTRONIC NUMERICAL INTEGRATOR AND COMPUTER. IT WAS BUILT BY A GROUP OF RESEARCHERS LED BY AMERICAN PHYSICIST JOHN MAUCHLY AT THE UNIVERSITY OF PENNSYLVANIA. UNLIKE EARLIER COMPUTERS, IT RAN ON RADIO TUBES—18,000 OF THEM IN TOTAL IT FILLED A ROOM 30 FEET BY 50 FEET AND COST \$400,000. USING TEN—DIGIT NUMBERS, IT COULD DO 5,000 ADDITIONS A SECOND. #. SP# A GERMAN PHYSICIST, ROENTGEN, DISCOVERED THE XRAY BY ACCIDENT. HE WAS DOING EXPERIMENTS WITH A CROOKES TUBE, WHICH PRODUCED STREAMS OF ELECTRONS CALLED CATHODE RAYS.

图7—3 power typing方式(第1部分)

INVENTOR SCRIPT AT V 132 TRUNC=132 SIZE=14 LINE=9 COLUMN=1

===== TOP OF FILE***

===== THE WORLD'S FIRST ELECTRONIC COMPUTER WAS CALLED ENIAC, ELECTRONIC
===== NUMERICAL INTEGRATOR AND COMPUTER. IT WAS BUILT BY AMERICAN PHYSICIST

===== JOHN MAUCHLY AT THE UNIVERSITY OF PENNSYLVANIA, UNLIKE EARLIER

===== COMPUTERS, IT RAN ON RADIO TUBES—18,000 OF THEM IN TOTAL, IT FILLED A

===== ROOM 30 FEET BY 50 FEET AND COST \$400,000, USING TEN—DIGIT NUMBERS, IT

===== COULD DO 5,000 ADDITIONS A SECOND.

===== .SP

===== A GERMAN PHYSICIST, ROENTGEN, DISCOVERED THE XRAY BY ACCIDENT, HE WAS
===== DOING EXPERIMENTS WITH A CROOKES TUBE, WHICH PRODUCED STREAMS OF

1....+....1....+....2....+....3....+....4....+....5....+....6....+....7....

===== ELECTRONS CALLED CATHODE RAYS.

===== END OF FILE***

=====

=====

===== ^

XEDIT 1 FILE

图7—3 power typing (第2部分)

3.1 行的分割和连接

PF10和PF11是用在文本处理中的二个键，它们分别在光标位置剖分和连接行

3.1.1 行的剖分 (PF10)

为了把一行分为二行，把光标移到想要另起一行的字符下并按PF10键。

在下行中，注意光标在“FOOD”的“F”下。

```
=====GILA MONSTERS HOLD RESERVE FOOD SUPPLIES IN THEIR TA  
ILS
```

按PF10键得到下面二行

```
=====GILA MONSTERS HOLD RESERVE_  
=====FOOD SUPPLIES IN THEIR TAILS
```

若要给一行加数据，PF10键特别有用。在下行中，光标放在“IN”中的“I”之下：

```
=====BIRD SPECIES HAVE DWINDLED IN THE LAST 70 MILLION Y  
EARS
```

按PF10键时它被分成二行：

```
=====BIRD SPECIES HAVE DWINDLED_  
=====IN THE LAST 70 MILLION YEARS
```

现在就有了在该行上加数据的地方：

```
=====BIRD SPECIES HAVE DWINDLED FROM 1.5 MILLION TO 10,000  
=====IN THE LAST 70 MILLION YEARS
```

3.1.2 二行的连接 (PF11)

按PF11键使二行在光标位置连起来。例如：

```
=====These lines are _  
=====too short
```

注意上面的光标位置。按PF11键得到下行。

```
=====These line are too short
```

记住：附加的或连上行会复盖光标后面的数据。例如：

```
=====The phrase _ "Things get" worse under pressure"  
=====is to be deleted
```

按PF11键复盖掉第一行光标后面的数据，结果为：

```
=====The phrase is to be deleted
```

3.2 屏幕的前后翻转

文件太长一屏放不下时，可以用PF7和PF8键向前或向后翻转文件。

按PF7键（它定为BACKWARD子命令）向文件顶部后翻一屏。反之，按PF8键（它定为FORWARD子命令）向文件尾部前翻一屏。反复按其中一键便可向前或向后翻很多屏。

3.3 子命令的再显示

在命令行打入的子命令执行后命令行清掉。如果使用PF键，子命令根本不在命令行中显示。若按错了PF键，或者未按想要的方式打入子命令之后想看执行的最后一个子命令，则可以按PF6键（它定为？子命令）在命令行中显示执行的最后一个子命令。

然后可以通过按ENTER键重新执行该子命令。如果子命令不正确，可以改正命令行上显示的错误然后按ENTER键。