

窗口中击键)。最起码, 键盘驱动程序必须解决在什么地方放置新键入的字符而不被程序的输出所覆盖。

当超过80个字符必须在具有80字符行(或某个其他数字)的窗口中显示时, 也使回显变得错综复杂。根据应用程序, 折行到下一行可能是适宜的。某些驱动程序只是通过丢弃超出80列的所有字符而将每行截断到80个字符。

另一个问题是制表符的处理。通常由驱动程序来计算光标当前定位在什么位置, 它既要考虑程序的输出又要考虑回显的输出, 并且要计算要回显的正确空格个数。

现在我们讨论设备等效性问题。逻辑上, 在一个文本行的结尾, 人们需要一个回车和一个换行, 回车使光标移回到第一列, 换行使光标前进到下一行。要求用户在每一行的结尾键入回车和换行是不受欢迎的。这就要求驱动程序将输入转化成操作系统使用的格式。在UNIX中, ENTER键被转换成一个换行用于内部存储; 而在Windows中, 它被转换成一个回车跟随一个换行。

如果标准形式只是存储一个换行(UNIX约定), 那么回车(由Enter键造成)应该转换为换行。如果内部格式是存储两者(Windows约定), 那么驱动程序应该在得到回车时生成一个换行并且在得到换行时生成一个回车。不管是什么内部约定, 监视器可能要求换行和回车两者都回显, 以便正确地更新屏幕。在诸如大型计算机这样的多用户系统上, 不同的用户可能拥有不同类型的终端连接到大型计算机上, 这就要求键盘驱动程序将所有不同的回车/换行组合转换成内部系统标准并且安排好正确地实现回显。

在规范模式下操作时, 许多输入字符具有特殊的含义。图5-35显示出了POSIX要求的所有特殊字符。默认的是所有控制字符, 这些控制字符应该不与程序所使用的文本输入或代码相冲突, 但是除了最后两个以外所有字符都可以在程序的控制下修改。

字 符	POSIX名	注 释
CTRL-H	ERASE	退格一个字符
CTRL-U	KILL	擦除正在键入的整行
CTRL-V	LNEXT	按字面意义解释下一个字符
CTRL-S	STOP	停止输出
CTRL-Q	START	开始输出
DEL	INTR	中断进程 (SIGINT)
CTRL-\	QUIT	强制核心转储 (SIGQUIT)
CTRL-D	EOF	文件结尾
CTRL-M	CR	回车 (不可修改的)
CTRL-J	NL	换行 (不可修改的)

图5-35 在规范模式下特殊处理的字符

ERASE字符允许用户删除刚刚键入的字符。它通常是退格符(CTRL+H)。它并不添加到字符队列中, 而是从队列中删除前一个字符。它应该被回显为三个字符的序列, 即退格符、空格和退格符, 以便从屏幕上删除前一个字符。如果前一个字符是制表符, 那么删除它取决于当它被键入的时候是如何处理的。如果制表符直接展开成空格, 那么就需要某些额外的信息来决定后退多远。如果制表符本身被存放在输入队列中, 那么就可以将其删除并且重新输出整行。在大多数系统中, 退格只删除当前行上的字符, 不会删除回车并且后退到前一行。

当用户注意到正在键入的一行的开头有一个错误时, 擦除一整行并且从头再来常常比较方便。KILL字符擦除一整行。大多数系统使被擦除的行从屏幕上消失, 但是也有少数古老的系统回显该行并且加上一个回车和换行, 因为有些用户喜欢看到旧的一行。因此, 如何回显KILL是个人喜好问题。与ERASE一样, KILL通常也不可能从当前行进一步回退。当一个字符块被删除时, 如果使用了缓冲, 那么烦劳驱动程序将缓冲区退还给缓冲池可能值得做也可能不值得做。

有时ERASE或KILL字符必须作为普通的数据键入。LNEXT字符用作一个转义字符(escape character)。在UNIX中, CTRL+V是默认的转义字符。例如, 更加古老的UNIX系统常常使用@作为KILL字符, 但是因特网邮件系统使用linda@cs.washington.edu形式的地址。有的人觉得老式的约定更加舒服从而将KILL重定义为@, 但是之后又需要按字面意义键入一个@符号到电子邮件地址中。这可以通过键入CTRL+V @来实现。CTRL+V本身可以通过键入CTRL+V CTRL+V而按字面意义键入。看到一个CTRL+V之后, 驱动程序设置一个标志, 表示下一字符免除特殊处理。LNEXT字符本身并不进入字符队列。

为了让用户阻止屏幕图像滚动出视线, 提供了控制码以便冻结屏幕并且之后重新开始滚动。在UNIX系统中, 这些控制码分别是STOP(CTRL+S)和START(CTRL+Q)。它们并不被存储, 只是用来设置或清除键盘数据结构中的一个标志。每当试图输出时, 就检查这个标志。如果标志已设置, 则不