

端类型，每一种都有自己的转义序列。因而，编写在一种以上的终端类型上工作的软件是十分困难的。

一种解决方案是称为termcap的终端数据库，它是在伯克利UNIX中引入的。该软件包定义了许多基本动作，例如将光标移动到（行，列）。为了将光标移动到一个特殊的位置，软件（如一个编辑器）使用一个一般的转义序列，然后该转义序列被转换成将要被执行写操作的终端的实际转义序列。以这种方式，该编辑器就可以工作在任何具有termcap数据库入口的终端上。许多UNIX软件仍然以这种方式工作，即使在个人计算机上。

逐渐地，业界看到了转义序列标准化的需要，所以就开发了一个ANSI标准。图5-36所示为一些该标准的取值。

下面考虑文本编辑器怎样使用这些转义序列。假设用户键入了一条命令指示编辑器完全删除第3行，然后封闭第2行和第4行之间的间隙。编辑器可以通过串行线向终端发送如下的转义序列：

ESC [3;1H ESC [0K ESC [1M

（其中在上面使用的空格只是为了分开符号，它们并不传送）。这一序列将光标移动到第3行的开头，擦除整个一行，然后删除现在的空行，使从第4行开始的所有行向上移动一行。现在，第4行变成了第3行，第5行变成了第4行，以此类推。类似的转义序列可以用来在显示器的中间添加文本。字和字符可以以类似的方式添加或删除。

转义序列	含 义
ESC [nA	向上移动n行
ESC [nB	向下移动n行
ESC [nC	向右移动n个间隔
ESC [nD	向左移动n个间隔
ESC [m;nH	将光标移动到 (m, n)
ESC [sJ	从光标清除屏幕 (0到结尾、1从开始、2两者)
ESC [sK	从光标清除行 (0到结尾、1从开始、2两者)
ESC [nL	在光标处插入n行
ESC [nM	在光标处删除n行
ESC [nP	在光标处删除n个字符
ESC [n@	在光标处插入n个字符
ESC [nm	允许再现n (0=常规、4=粗体、5=闪烁、7=反白)
ESC M	如果光标在顶行上则向后滚动屏幕

图5-36 终端驱动程序在输出时接受的ANSI转义序列。ESC表示ASCII转义字符(0x1B)，n、m和s是可选的数值参数

## 2. X窗口系统

几乎所有UNIX系统的用户界面都以X窗口系统(X Window System)为基础，X窗口系统经常仅称为X，它是作为Athena计划<sup>①</sup>的一部分于20世纪80年代在MIT开发的。X窗口系统具有非常好的可移植性，并且完全运行在用户空间中。人们最初打算将其用于将大量的远程用户终端与中央计算服务器相连接，所以它在逻辑上分成客户软件和主机软件，这样就有可能运行在不同的计算机上。在现代个人计算机上，两部分可以运行在相同的机器上。在Linux系统上，流行的Gnome和KDE桌面环境就运行在X之上。

当X在一台机器上运行时，从键盘或鼠标采集输入并且将输出写到屏幕上的软件称为X服务器(X server)。它必须跟踪当前选择了哪个窗口(鼠标指针所在处)，这样它就知道将新的键盘输入发送给哪个客户。它与称为X客户(X client)的运行中的程序进行通信(可能通过网络)。它将键盘与鼠标输入发送给X客户，并且从X客户接收显示命令。

① Athena(雅典娜)指麻省理工学院(MIT)校园范围内基于UNIX的计算环境。——译者注