

第24章 交互式终端

本章及下一章的任务是分析控制交互式终端的代码(或简称为“终端”)。

有很多种终端可供选择使用,而在一台计算机上可能同时连接若干种不同类型的终端。

区分不同类型终端的主要特征包括(除形状、大小和色彩等非本质的区别之外):

- 1) 传输速度:例如 ASR33电传打印机的110波特率;DECwriter的300波特率,视频单元(VDU)的2400或9600波特率。
- 2) 图形字符集:全ASCII图形集和64图形子集。
- 3) 传输的奇、偶校验:奇校、偶校或者无奇偶校。
- 4) 输出技术:串行打印机或视频显示。
- 5) 其他:组合的回车/换行字符;半双工终端(输入字符无需回打);制表符的识别。
- 6) 对于某些控制功能的延迟特征:例如在传输1个字符所需时间内不能完成回车等动作。

24.1 接口

与有大量不同类型的终端可用并正在使用一样,也有很多种硬件设备可用作为终端和PDP11计算机之间的接口。例如:

- DL11/KL11 单线、异步接口;40~9600波特率之间的13种标准传输速率。
- DJ11 16线、异步、带缓存的串行线多路转接器;在4线组中,75~9600波特率之间可选择的11种速度。
- DH11 16线、异步、带缓存的串行线多路转接器;单个、可选的、14种速度。

上列各种接口都可以用全或半双工方式工作;处理5、6、7或8级代码;产生奇、偶或无校验;产生1、1.5或2位的暂停码。

除了上列异步接口外,也有一定数量的同步接口,例如DQ11。

每一接口都有其自己的控制特征,它要求一单独的操作系统设备驱动程序。这些驱动程序之间可共享的代码集中在一个文件“tty.c”中。一套公共定义集中在文件“ttg.h”中。

作为样本,包含了文件“kl.c”,其中的代码构成了DL11/KL11接口集的设备驱动程序。因为一台KL11作为系统操作员的控制终端是系统的必备设备,所以该驱动程序也是必需提供的。

24.2 tty结构(7926)

每一个终端端口都需要一个“tty”型结构与其关连(不论使用的是哪一种硬件接口)。在本文中,“端口”指的是连接终端线的一个位置。因此一个DL11只提供一个端口,而一个DJ11则提供多至16个端口。

“tty”结构由16个字组成,如下所示:

- | | | |
|---|----------|--|
| A | t_dev | 对一特定终端口设置成固定值 |
| | t_addr | |
| B | t_speeds | 对一特定终端口设置成固定值。这些值可由“stty”设置，可由“gtty”查询 |
| | t_erase | |
| | t_kill | |
| | t_flags | |
| C | t_rawq | 3个字符队列的列表头，它们分别是“原始”输入(raw input)、非原始输入(cooked input)和输出队列 |
| | t_canq | |
| | t_outq | |
| D | t_state | 状态信息，在通常的处理中这些信息频频发生变化 |
| | t_delct | |
| | t_col | |
| | t_char | |

24.3 注释

下面列出的一些条目在本章选择分析的代码中都基本上没有涉及到：

| | | | |
|----------|--------|---------|--------|
| t_char | (7940) | NLDELAY | (7974) |
| t_speeds | (7941) | TBDELAY | (7975) |
| | | AY | (7976) |
| | | [| (7985) |
| | | , | (7993) |

动程序中的“open”例程例如“klopen”(8023)实现的。
y”系统调用更改。它们的当前值则可由“gtty”系统调用

UPM的STTY()和GTTY()部分。它们由shell命令
|请参阅UPM的STTY()部分。

用要求一个文件描述符作为其参数，所以只能将它们用于

代码。下面我们将跟踪“stty”的执行过程。而对“gtty”

它使用指针参数将3个字的用户参数信息复制到“u.u_arg

24.6 sgty(8201)

8206：取得一指向“file”数组项的有效指针。

8209：检查该文件是一个“字符特殊”文件。

8213：对该设备类型调用合适的“d_sgty”例程。

注意，对于行式打印机和纸带阅读机/穿孔机“d_sgty”例程是“nodev”。

24.7 klsqty(8090)

这是“d_sgty”例程的一个实例。它调用“tlysty”，调用参数是指向适当的“tty”结构的指针。

24.8 ttysty(8577)

对本过程调用起始于“sty”，其第2个参数值为0。

8589：将与此终端相关的所有队列都立即清除为空。很可能它们包含的信息都是毫无意义的。

8591：复位速度信息(对于DH11接口，这是有用的，但是对于我们现在所选择的代码却并无意义)。

8592：复位“擦除”(erase)字符和“擦除行”(kill)字符。(kill在这里表示“丢弃当前输入行”。)注意，假如这些字符分别改变为非正常值(它们的正常值为#和@)，对“maptab”则不作相应更改。

8593：复位定义某些相关终端特征的标志：

| 标 志 | 位 | 如 果 设 置..... |
|-------|---|--|
| XTABS | 1 | 此终端将不正确地解释水平制表符 |
| LCASE | 2 | 此终端只支持64字符ASCII子集 |
| ECHO | 3 | 此终端以全双工方式运作，输入字符应回打 |
| CRMOD | 4 | 在输入中，将“回车”代换成“换行”；在输出中，将“换行”代换成“回车”和“换行” |
| RAW | 5 | 将输入字符按原样送给程序，不进行“擦除”和“擦除行”处理，或对于反斜线字符的调整 |

另外，下列位由“ttyoutput”(8373)查询，以选择发送所指明的字符后及发送下一个字符之前应当有的延迟时间：

- 8, 9 换行
- 10, 11 水平制表
- 12, 13 回车
- 14 垂直制表或换页

24.9 DL11/KL11终端设备处理程序

文件“kl.c”包含经DL11/KL11连接至系统的各终端设备处理程序。这组终端至少有一个

成员——操作员控制终端，因此这一设备处理程序总是存在于系统中。

每一DL11/KL11硬件控制器提供一个异步串行接口，它将一个终端连接至一台 PDP11系统。关于此接口的更完整细节，请读者参阅《PDP11 Peripherals Handbook》。

24.10 设备寄存器

每一DL11/KL11单元有4个寄存器，它们在单总线上占用 4个连续字。UNIX将一类型为“kregs”的结构(8016)映照至一个这样的寄存器组。

24.11 接收器状态寄存器

第7位 接收器完成。(已将一字符传输至接收器数据缓存寄存器。)

第6位 接收器中断允许。(当设置时，每次第7位被设置时就造成一次中断。)

第1位 数据终端准备就绪。

第0位 阅读器使能。只写。(当设置时，第7位被清除。)

24.12 接收器数据缓存寄存器

第15位 当设置时表示已出错。

第7~0位 接收到的字符，只读。

24.13 发送器状态寄存器

第7位 发送器准备就绪。当数据装入发送器数据缓存时，此位被清除；当后者准备好

时，只要第7位被设置，就产生一中断。)

。

是从4字边界处开始(下面的地址都是18位八进制地址。)

| 态 | | 发送器状态 |
|---|-----|--------|
| 0 | - > | 777566 |
| 0 | - > | 776506 |
| 0 | - > | 776516 |
| 0 | - > | 776676 |
| 0 | - > | 775616 |
| 0 | - > | 775626 |
| 0 | - > | 776176 |

标准总线单元，除此以外其他接口被分成两组(作这种划

分的理由此处不再展开)。在每一组中,按约定,为各寄存器分配在最低地址开始的连续单元。

24.16 软件方面的考虑

应当设置“NKL11”(8011)以便为一特定装置规定在前2个组中的接口数,也应设置“NDL11”(8012)以便为该装置规定在第3个组中的接口数。如果在硬件方面对实际的接口数作了改变,那么就应将这种改变反映到软件中,方法是对“kl.c”作相应更改,然后重新编译,最后重新连接操作系统。

我们将会了解到,“klopen”为每个接口的接收器状态寄存器计算正确的核心态地址(16位),然后将其存放在相应“tty”结构的“t_addr”元素中(8044)。

24.17 中断矢量地址

第1个接口的中断矢量地址是060和064(分别对于接收器和发送器接口)。另外的DL11/KL11接口中断矢量地址最低为0300,并按一定规则对它们进行分配,在此规则中考虑到了可能存在的其他接口。

中断矢量的第2个字是“新处理机状态”字。该字的最低5位值可以任意选择,事实上被用于规定次设备号(在陷入矢量中也有类似用法,以区分不同类型的陷入)。

经屏蔽处理后的新处理机状态字作为参数dev被提供给中断处理例程(8070行)。

24.18 源代码

现在我们可以转而详细分析在文件“kl.c”和“tty.c”中的源代码。我们首先来观察的是如何“打开”和“关闭”作为字符特殊文件的终端,以及一章我们将分析如何从终端接收数据以及如何向终端发送数据。

“klread”(8062)、“klwrite”(8066)和“klsgtty”(8090不再重复。

24.19 klopen(8023)

调用此过程以作为一字符特殊文件打开一终端。“/etc/”的终端调用此过程。因为子进程继承其父进程的所有打开要再打开终端设备。我们将会注意到,即使两个无关的进程打开文件,UNIX也无意停止它们。

8026:检查次设备号。

8030:定位相应的“tty”结构。

8031:如果打开此文件的进程并无相关连的控制终端存放至该进程p_tty项的是一个tty结构的地址。)

8033:在“tty”结构中存入终端设备号。

8039:为该终端计算相应设备寄存器组的地址,并将其

8045:若该终端尚未打开,则对该“tty”结构进行某

8046：设置“t_state”以表示该文件已打开，这样就使得若以后再打开此文件，那么后面3行就不会被执行，可能去除了一条“stty”系统调用的作用。

“t_state”也被设置“CARR_ON”。这是一个软件标志，它表示该终端在逻辑上已起作用，这与终端硬件的真实状态无关。若某个终端的“CARR_ON”标志为0，那么系统应当忽略来自该终端的所有输入。（似乎并不完全如此，我们将会在后面再次提及此点。）

8047：系统认为标准终端是：不能解释水平制表符；只支持64字符的ASCII子集，以全双工方式运行；要求“回车”和“换行”字符以提供正常的“新行”处理。（这种标准终端是Model 33电传打字机？）

8048：按UNIX约定设置“擦除”和“擦除行”字符。

8051：用“0103”模式对接收器状态寄存器设置初值，使终端准备就绪，可以读该终端，并开启接收器中断。

8052：对发送器控制状态寄存器赋初值，使得当接口为接收另一个字符准备就绪时能够产生一次中断。

注意，“open”例程并不对该文件为只读、只写或读/写打开等不同情况进行区分。

24.20 kclose(8055)

8057：在“tty”结构数组“kl11”（8015）中找到适当的“tty”结构的地址。

8058：“wflushtty”（8217）使该终端的输出队列中的全部字符皆输出，然后刷清输入队列。

8059：复位“t_state”，清除“ISOPEN”和“CARR_ON”标志。

程。应该将此过程与“pcpint”（8739）和“lpint”（8976）中断矢量中的“新处理机状态”字经屏蔽处理过的（保存了初值，那么该次设备号也应当是正确的。一章的结尾处讨论。

程。虽然与“pccint”（8719）相比较有一些相似之处，但

中读输入字符。

！。

硬件把工作搞坏了）。最好相信这一注释。

it”，它将该字符插入合适的“原始”输入队列。

