

POSIX 操作系统 串行通讯 编程指南



POSIX 操作系统 串行通讯 编程指南

5th Edition, 6th Revision
Copyright 1994-2005 by Michael R. Sweet

Translated by Cedar Peng (翻译 彭雪松)

允许基于 [GNU Free Document License](#) 1.2 版本（或任何由自由软件基金会发布的更新版本）下的复制、分发或修改，但不改变章节，不增加额外开头和结尾文字说明。附录 C，包含一份 **GNU Free Documentation License** 的副本。

目录

介绍

- 许可
- 组织

第一章，串行通讯编程基础

- 什么是串行通讯？
- 什么是 **RS-232**？
 - 信号定义
- 异步通讯
 - 什么是全双工和半双工？
 - 流控制
 - 什么是中断？
- 同步通讯
- 访问串口
 - 串口文件
 - 打开串口
 - 向端口写数据
 - 从端口读数据
 - 关闭串口

第二章，设置串口

- **POSIX** 终端接口
 - 控制项
 - 本地项
 - 输入项
 - 输出项
 - 控制字符

第三章，**MODEM** 通讯

- 什么是 **MODEM**？
- 与 **MODEM** 通讯
 - 标准 **MODEM** 指令
 - 常见 **MODEM** 通讯问题

第四章，高级串行通讯编程

- 串行端口 **IOCTLs**
 - 获得控制信号
 - 设置控制信号
 - 获得有效的字节数目
- 从串行端口选择输入
 - **SELECT** 系统调用
 - 使用 **SELECT** 系统调用
 - 借助 **X** 系统内置库，使用 **SELECT**

附录 A，引脚定义

- **RS-232** 引脚定义
- **RS-422** 引脚定义
- **RS-574 (IBM PC/AT)** 引脚定义
- **SGI** 引脚定义

附录 B，ASCII 控制代码

- 控制代码

附录 C，GNU Free Documentation License

- 如何为你的文件使用该许可

附录 D，修改历史

- **Edition 5, Revision 6**
- **Edition 5, Revision 5**
- **Edition 5, Revision 4**
- **Edition 5, Revision 3**

介绍

POSIX 操作系统 串行通讯 编程指南 将教会你如何成功、有效和可移植地，在你的 UNIX 工作站或 PC 上，进行串口编程。每章提供的编程样例，采用 POSIX (UNIX 可移植标准) 终端控制功能，少量修改后就应可以运行于 IRIX®, HP-UX, SunOS®, Solaris®, Digital UNIX®, Linux®, 和多数其它 UNIX 操作系统。你将发现在不同操作系统之间的最大差别，其实是串口设备和锁定文件的**名字**。

许可

允许基于 [GNU Free Document License](#) 1.2 版本（或任何由自由软件基金会发布的更新版本）下的复制、分发或修改，但不改变章节，不增加额外开头和结尾文字说明。附录 C，包含一份 **GNU Free Documentation License** 的副本。

组织

这篇指南由以下章节和附录组成：

- ◆ **第一章，串行通讯编程基础**
- ◆ **第二章，配置串口**
- ◆ **第三章，与 MODEM 通讯**
- ◆ **第四章，高级串行通讯编程**
- ◆ **附录 A，引脚定义**
- ◆ **附录 B，ASCII 控制码**
- ◆ **附录 C，GNU Free Documentation License**
- ◆ **附录 D，修改历史**

第一章，串行通讯编程基础

这章介绍了串行通讯，RS-232 和其它一些在计算机上广泛使用的标准，以及如何从 C 程序访问串口。

什么是串行通讯？

计算机一次传输一位或多位的信息（数据）。串行指一次传输一位数据。串行通讯包括了大多数的网络设备、键盘、鼠标、MODEM 和终端。

当进行串行通讯时，你传送或接受的每个词（例如：字节或字符）是以每次一位传输的。每位或者为 *on* 或者 *off*。有时你也能听到以 *mark* 表示 *on* 状态和 *space* 表示 *off* 状态。

串行数据的速度常被表示为比特每秒 ("bps") 或波特率 ("波特")。这是用于代表每秒能够传送的 1 和 0 的数量。追溯到计算机时代的早期，300 波特曾被认为是很快的速度，但今天的计算机能够控制 RS-232 速度高达 430,800 波特！当波特率超过 1,000，你将常看到速度被表示为 *kilo* 波特，或 kbps（例如 9.6k, 19.2k 等等）。对于超过 1,000,000 的速度表示为 *mega* 波特，或者 Mbps（例如 1.5Mbps）。

当提及串行设备或端口的时候，他们总被标明数据通讯设备 ("DCE")或者数据终端设备 ("DTE")。他们之间的差别很简单 -- 每个信号对，例如传送和接受，都需要交换。而当连接两个 DTE 或两个 DCE 接口到一起的时候，需要使用一个串行 *null-MODEM* 缆线或适配卡来交换信号对。

什么是 RS-232？

RS-232 是标准的串行通讯电气接口，由 **Electronic Industries Association ("EIA")**定义。RS-232 实际有三个不同类型(A, B 和 C)，每类都定义了不同的 *on* 与 *off* 的电压范围。最为常用的是 **RS-232C**，它定义 *mark* (*on*) 的电压在-3V 到-12V 之间和 *space* (*off*)的电压+3V 到 +12V 之间。RS-232C 式样规定这些信号 25 英尺 (8 米)内有效。通常只要波特率足够低，你能够传送更长的距离。

除了输入和输出数据的引线，还有提供计时、状态和握手信号的引线：

表 1 - RS-232 引脚定义

PIN	描述	PIN	描述	PIN	描述	PIN	描述	PIN	描述
1	接地	6	DSR -Data Set Ready	11	未定义	16	第 二 个 RXD	21	信号质量检测
2	TXD -Transmitted Data	7	GND -逻辑地	12	第 二 个 DCD	17	Receiver 时钟	22	响铃检测
3	RXD -Received Data	8	DCD -Data Carrier Detect	13	第 二 个 CTS	18	未定义	23	Data Rate Select
4	RTS -Request To Send	9	保留	14	第 二 个 TXD	19	第 二 个 RTS	24	Transmit 时钟

5	CTS- Clear To Send	10	保留	15	Transmit 时钟	20	DTR Data Terminal Ready	-	25	未定义
---	---------------------------------	----	----	----	----------------	----	-----------------------------------------	---	----	-----

其它两个串口标准，你可能也看到过，是 RS-422 和 RS-574。RS-422 使用低电压和微分信号，允许线缆长度到 1000 英尺 (300 米)。RS-574 定义了 9-pin PC 串行接口及其电压。

信号定义

RS-232 标准为串行通讯定义了 18 种不同的相关信号。然而，UNIX 环境下只有 6 种有效。

GND – 逻辑地

就技术而言，逻辑地不是个信号，但没有它，其它信号都不能工作。简而言之，逻辑地作为参考电压，从而使电子器件知道某个电压是正或负。

TXD – 数据发送

TXD 信号负责从你的工作站将数据传送到另一端的计算机或设备(例如 MODEM)。一个 mark 电压被解释为 1，与此同时一个 space 电压被解释为 0。

RXD – 数据接收

RXD 信号负责接收另一端的计算机或设备传送到你的工作站上的数据。类似 TXD，mark 和 space 电压相应地被解释为 1 和 0。

DCD – 载波侦听

DCD 信号负责接受从串行电缆另一头的计算机或其它设备，传送过来的信号。一个 space 电压表示计算机或设备当前连接着或在线。DCD 并不被使用或有效。

DTR - Data Terminal Ready

DTR 信号由你的工作站产生，告诉另一端的计算机或设备你准备好了 (space 电压) 或者没有准备好 (mark 电压)。当你在工作站上打开串行接口时，DTR 通常是自动起效的。

CTS - Clear To Send

CTS 信号来自串行线缆的另一端。一个 space 电压表示你可以从工作站发送更多的数据。

CTS 通常被用作控制从你的工作站发往另一端的串行数据流。

RTS - Request To Send

RTS 信号被由你的工作站设置为 space 电压，表示还有数据等待传送。

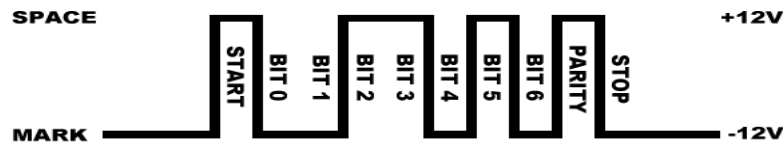
类似于 CTS，RTS 帮助控制你的工作站和另一端的计算机或设备间的数据流。多数工作站会始终保持这个信号为 space 电压。

异步通讯

为使计算机理解它收到的串行数据，需要一些途径决定字符的 *开始* 和 *结束*。本指南特别讨论异步串行数据。

在异步模式下，串行数据线保持在 *mark*(1) 状态，直到有字符传送。起始是一比特 *start* 位，接着字符内容的每一位比特，然后是一比特可选的校验位，最后由一比特或更多的 *stop* 位。起始位始终是一个 *space* (0)，从而告诉计算机新的串行数据到来。数据能够在任何时候传送或接受，所以称为异步。

图 1 - 异步数据传输



可选的校验位是一个简单数据位的加和，表明了数据位包含偶数或奇数个比特 1。如果是偶校验，当在字符中的 1 是偶数个数时，校验位为 0。如果是奇校验，当字符中的 1 是奇数个数时，校验位为 0。你可能也听说过 *space* 校验，*mark* 校验，和无校验。*Space* 校验意味着校验位始终为 0，而 *mark* 校验意味着校验位始终为 1。无校验意味着校验位不存在或不被传输。

余下的被称为停止位。这可以是 1, 1.5, 或 2 位停止位在字符之间，并始终是 1 值。停止位原先是用于给计算机时间处理前面字符的，但现在只是用来帮助接收计算机同步输入字符。

异步数据格式通常表示为 "8N1", "7E1", 诸如此类。它们相应地表示 "8 位数据，无校验，1 位停止" 和 "7 位数据，偶校验，1 位停止"。

什么是全双工与半双工？

全双工 表明计算机能够同时接受和发送数据 - 有两个分离的数据通道（一个输入，一个输出）。

半双工 表明计算机不能同时接受和发送数据。通常这意味着只有一个数据通道用来通讯。这并不意味着一些 RS-232 信号没有使用。而通常是表明通讯链路使用了不同于 RS-232 的标准，而它们不支持全双工操作。

流量控制

通常在两个串口间传送数据时，需要控制数据流。这可能是由于串行通讯链接、其中一个串行接口、或存储介质的限制。有两种方法常用于异步数据。

第一个方法是 "软件" 流量控制，使用特殊字符开始 (XON or DC1, 021 octal) 或停止 (XOFF or DC3, 023 octal) 数据流。这些字符由 **American Standard Code for Information Interchange ("ASCII")** 定义。虽然这些代码在传送文本信息时十分有用，但没有特殊的编程手段，在传输其它类型信息的时候就不能使用了。

第二个方法叫做 "硬件" 流量控制，使用 RS-232 CTS 和 RTS 信号代替特殊字符。当准备好接受更多的数据后，接受方设置 CTS 为 *space* 电压；而设置为 *mark* 电压，表示没有准备好。类似地，准备好发送更多数据后，发送方将 RTS 设置为 *space* 电压。因为硬件流量控制使用独立的信号，所以同样条

件下，它要比需要传送更多位信息的软件流量控制快许多。然而 CTS/RTS 流量控制并不被所有的硬件或操作系统支持。

什么是中断？

通常的一个接受或发送数据信号线保持在 mark 电压，直到要传送一个新的字符。如果信号跌落到 space 电压并持续一段时间，通常是 1/4 到 1/2 秒，就认为是出现了个中断。

中断有时是用于重置通讯线 或改变通讯硬件的操作模式，例如 MODEM.第三章，MODEM 通讯更为深入地讨论了这些。

同步通讯

不同于异步数据，同步数据表现为一个持续比特流。为读取在线数据，计算机必须提供或接受同一同步比特时钟，从而使接受与发送同步。

即便是同步情况下，计算机仍然必须标明数据的开始。最通常的做法是使用数据包协议，例如串行数据链路控制 Serial Data Link Control ("SDLC")或 High-Speed Data Link Control ("HDLC")。

每个协议都定义了表示数据包开始和结束的特定比特序列，并且都定义了没有数据时的比特序列。这些比特序列使计算机能够发现数据包的起始位置。

因为同步协议不在每个字符中使用同步位，它们通常提供比异步通讯至少高 25%的效率，适合于在两个以上的串行接口环境下，远距离的网络操作与配置。

尽管同步通讯的速度优势明显，多数 RS-232 硬件仍不支持，主要是由于额外的硬件和软件需求。

访问串行端口

与所有设备一样，UNIX 提供 设备文件 以访问串行端口。要访问一个串行端口，你只要打开相应的设备文件即可。

串行端口文件

每个串行端口在 UNIX 系统上有一个或多个设备文件（在/dev 目录下）相关联：

表 2 - 串行端口设备文件		
系统	端口 1	端口 2
IRIX®	/dev/ttyf1	/dev/ttyf2
HP-UX	/dev/tty1p0	/dev/tty2p0
Solaris®/SunOS®	/dev/ttya	/dev/ttyb
Linux®	/dev/ttyS0	/dev/ttyS1
Digital UNIX®	/dev/tty01	/dev/tty02

打开串行端口

由于串行端口是一个文件，open(2) 函数被用于访问它。有一个障碍是，UNIX 下的设备文件通常普通用户没有访问权限。解决的办法包括，改变提及文件访问权限，以超级用户(root)运行你的程序，或使你

的程序能设置 `userid`，从而它能以设备文件的拥有者身份运行(由于显著的安全性问题，不建议这么做...)

现在，我们假设文件能够被所有的用户访问。**列表 1** 表示了在运行 Linux 的 PC 上打开串行端口一的例子。

列表 1 - 打开串行端口.

```
#include <stdio.h> /* 标准输入、输出定义 */
#include <string.h> /* 字符串函数定义 */
#include <unistd.h> /* UNIX 标准函数定义 */
#include <fcntl.h> /* 文件控制定义 */
#include <errno.h> /* 错误编号定义 */
#include <termios.h> /* POSIX 终端控制定义 */

/*
 * 'open_port()' - 打开串行端口 1
 *
 * 成功后返回文件描述符，或是失败后返回-1。
 */

int
open_port(void)
{
    int fd; /* 端口文件描述符 */

    fd = open("/dev/ttyS0", O_RDWR | O_NOCTTY | O_NDELAY);
    if (fd == -1)
    {
        /*
         * 打开端口失败
         */

        perror("open_port: Unable to open /dev/ttyS0 - ");
    }
    else
        fcntl(fd, F_SETFL, 0);

    return (fd);
}
```

其它系统可能需要不同的对应设备文件名，然而代码是一样的。

Open 选项

你会注意到我们打开设备文件时，除了读写模式外，还使用了两个其它的标志：

```
fd = open("/dev/ttyS0", O_RDWR | O_NOCTTY | O_NDELAY);
```

`O_NOCTTY` 标志告诉 UNIX，该程序不是那个端口的"控制终端"。如果你不指明这个，任何输入(例如键盘中止信号，诸如此类)都会影响你的进程。类似 *getty (1M/8)* 之类的程序在启动登录进程时，设置这一特性，但通常用户的程序不希望这种行为。

`O_NDELAY` 标志告诉 UNIX，该程序不关心 DCD 信号的状态 - 即无论另一端是否启用和运行。如果你不指明这个标志，你的进程将休眠，直到 DCD 信号线是 space 电压。

向端口写数据

向端口写数据是很简单的 - 只要使用 `write(2)` 系统调用来发送数据即可：

```
n = write(fd, "ATZ\r", 4);
if (n < 0)
    fputs("write() of 4 bytes failed!\n", stderr);
```

`write` 函数返回发送的字节数目，如果出错就返回-1。通常你运行时唯一可能进入的错误是 `EIO`，即当 MODEM 或数据链路的数据载波侦听(DCD) 线失效。该状况将一直保持，直到你关闭端口。

从端口读数据

从端口读取数据有一个小技巧。当你以 `raw` 数据模式操作端口时，`read(2)` 系统调用都会返回当前串行输入缓冲中实际有效的字符数目。如果没有字符存在，调用将阻塞(等待)直到有字符进入，超时，或出错。以下设置，能使得 `read` 方法立即返回：

```
fcntl(fd, F_SETFL, FNDELAY);
```

当端口没有接受到字符时，`FNDELAY` 项使得 `read` 函数返回 0。要重新实现标准的(阻塞)行为，调用 `fcntl()` 而不带 `FNDELAY` 项即可：

```
fcntl(fd, F_SETFL, 0);
```

这同样可在以 `O_NDELAY` 选项打开串行端口后使用。

关闭串口

要关闭串口，使用 `close` 系统调用即可：

```
close(fd);
```

关闭一个串口，通常同时会将 DTR 信号设置为低，这将使多数 MODEM 挂机。

第二章，设置串行端口

本章讨论如何使用 POSIX `termios` 接口，在 C 语言中设置端口。

POSIX 终端接口

多数系统支持通过 POSIX 终端（串行）接口来改变参数，例如波特率，字符尺寸，等等。你首先需要做的是引入文件 `<termios.h>`；它定义了终端控制结构和 POSIX 控制函数。

两个最重要的 POSIX 函数是 `tcgetattr(3)` 和 `tcsetattr(3)`。相应地，它们获得和设定相关的终端特性，你提供一个 `termios` 结构的指针，包含所有可用的串行设置项：

表 3 - Termios 结构成员	
设置项	描述
c_cflag	控制项
c_lflag	线路项
c_iflag	输入项
c_oflag	输出项
c_cc	控制字符
c_ispeed	输入波特(新接口)
c_ospeed	输出波特(新接口)

控制项

`c_cflag` 设置项控制波特率、数据位、校验、停止位和硬件流控制。所有支持的设置都有常量对应。

表 4 - c_cflag 成员常量	
常量	描述
CBAUD	波特率比特掩码
B0	0 波特 (降下 DTR)
B50	50 波特
B75	75 波特
B110	110 波特
B134	134.5 波特
B150	150 波特
B200	200 波特
B300	300 波特
B600	600 波特

B1200	1200 波特
B1800	1800 波特
B2400	2400 波特
B4800	4800 波特
B9600	9600 波特
B19200	19200 波特
B38400	38400 波特
B57600	57,600 波特
B76800	76,800 波特
B115200	115,200 波特
EXTA	外部速率时钟
EXTB	外部速率时钟
CSIZE	数据位的比特掩码
CS5	5 数据位
CS6	6 数据位
CS7	7 数据位
CS8	8 数据位
CSTOPB	2 停止位 (不指明是 1 位)
CREAD	接收有效
PARENB	校验有效
PARODD	使用奇校验, 代替偶校验
HUPCL	在最后一个关闭后, 挂起 (降下 DTR)
CLOCAL	本地线 - 不改变端口的拥有者
LOBLK	阻塞方式, 控制输出
CNEW_RTSCCTS CRTSCTS	采用硬件流控制 (在所有的平台上都不支持)

`c_cflag` 设置项有两个选项必须总是启用, `CLOCAL` 和 `CREAD`。这些将保证你的程序不会成为端口的‘所有者’, 从而妨碍控制工作或挂起信号, 并使串行接口驱动能读取收到的数据。

波特率常数 (`CBAUD`, `B9600`, 等等。)是用来兼容老式系统接口, 某些缺少 `c_ispeed` 和 `c_ospeed` 设置项的。参考下一章, 关于波特率设定的 POSIX 函数。

决不 直接初始化 `c_cflag` (或其它)设置项; 你应该总是用 `AND`, `OR` 和 `NOT` 比特操作来设定或清除设置项中的比特位。不同的操作系统版本(甚至是补丁)可能并确实用不同方式处理比特位, 所以使用比特操作符, 能够使你避免在操作一个新的串行设备比特位时遭遇意外。

设置波特率

基于不同的操作系统, 波特率存储于不同的位置。旧版本系统接口使用一个表 4 中的波特率常数, 将波特率存在 `c_cflag` 设置项中, 而新的实现提供了 `c_ispeed` 和 `c_ospeed` 设置项来保存波特率数值。

`cfsetospeed(3)` 和 `cfsetispeed(3)` 函数用于将波特率设置到 `termios` 结构中，而不论其下的操作系统接口如何。通常你使要用 [列表 2](#) 中编码来设置波特率。

列表 2 – 设置波特率.

```
struct termios options;

/*
 * 获得当前端口设置项...
 */

tcgetattr(fd, &options);

/*
 * 将波特率设置为 19200...
 */

cfsetispeed(&options, B19200);
cfsetospeed(&options, B19200);

/*
 * 启用接收项，并设置本地模式...
 */

options.c_cflag |= (CLOCAL | CREAD);

/*
 * 位端口设置新选项...
 */

tcsetattr(fd, TCSANOW, &options);
```

`tcgetattr(3)` 函数使用当前的串行端口设置，来填充 `termios` 结构。在我们设置了波特率、启用本地模式和串行数据接收，即可使用 `tcsetattr(3)` 写入新的设置项。`TCSANOW` 常量标明所有的改变立即生效，而不等待发送或接受的数据结束。也有其它的常量用于标明，等待输出和输入结束，或刷新输入输出缓冲。

多数系统不支持端口的输入与输出速率不同，所以为了最大的可移植性，请将它们设置为相同的值。

表 5 – tcsetattr 的常量	
常量	描述
TCSANOW	不等待数据完成，立即做出改变
TCSADRAIN	等待所有事情完成，再改变
TCSAFLUSH	刷新输入和输入缓冲，并作出改变

设置字符位尺寸

不同于波特率，没有简便的函数用于设定字符位尺寸。取而代之的是你需要做一点比特掩码的操作来实现。字符尺寸是以位多少设定：

```
options.c_cflag &= ~CSIZE; /* 标记字符尺寸位 */
```

```
options.c_cflag |= CS8; /* 选择 8 位数据位 */
```

设置位校验

类似字符尺寸，你必须手动设置位校验有效及位校验类型。UNIX 串行驱动通常支持偶，奇，和无校验产生。空位校验能够通过聪明的编码来模拟。

- 无校验位 (8N1):
 - `options.c_cflag &= ~PARENB`
 - `options.c_cflag &= ~CSTOPB`
 - `options.c_cflag &= ~CSIZE;`
 - `options.c_cflag |= CS8;`
- 偶校验 (7E1):
 - `options.c_cflag |= PARENB`
 - `options.c_cflag &= ~PARODD`
 - `options.c_cflag &= ~CSTOPB`
 - `options.c_cflag &= ~CSIZE;`
 - `options.c_cflag |= CS7;`
- 奇校验 (7O1):
 - `options.c_cflag |= PARENB`
 - `options.c_cflag |= PARODD`
 - `options.c_cflag &= ~CSTOPB`
 - `options.c_cflag &= ~CSIZE;`
 - `options.c_cflag |= CS7;`
- 空校验与无校验的设置相同 (7S1):
 - `options.c_cflag &= ~PARENB`
 - `options.c_cflag &= ~CSTOPB`
 - `options.c_cflag &= ~CSIZE;`
 - `options.c_cflag |= CS8;`

设置硬件流控制

某些版本的 UNIX 支持硬件流控制，采用的是 CTS (Clear To Send) 和 RTS (Request To Send) 信号线。如果 `CNEW_RTSCCTS` 或 `CRTSCCTS` 常量在你的系统上有定义，硬件流控制有可能被支持。以下用于启用硬件流控制：

```
options.c_cflag |= CNEW_RTSCCTS; /* 也称为 CRTSCCTS */
```

类似地，关闭硬件流控制：

```
options.c_cflag &= ~CNEW_RTSCCTS;
```

本地项

本地模式设置项 *c_lflag* 控制输入字符如何被串行驱动操作。一般你会设置 *c_lflag* 成员为 *canonical* 或 *raw* 输入方式。

表 6 - c_lflag 设置项常量	
常量	描述
ISIG	启用 SIGINTR, SIGSUSP, SIGDSUSP, and SIGQUIT 信号
ICANON	启用经典方式输入(否则就是 raw)
XCASE	Map uppercase \lowercase (过时)
ECHO	启用输入字符回显
ECHOE	回显擦除字符 BS-SP-BS
ECHOK	删除字符后，回显 NL
ECHONL	回显 NL
NOFLSH	中断或退出字符后，停止刷新输入缓冲
IEXTEN	启用扩展函数
ECHOCTL	将控制字符回显为 ^char，删除为 ~?
ECHOPRT	回显删除字符，就将字符删除掉
ECHOKE	在有行删除时候，BS-SP-BS 整行
FLUSHO	输出缓冲被刷新
PENDIN	在下次读取或输入字符时，重打印挂起的输入
TOSTOP	发送 SIGTTOU 作为输出背景

选择经典方式输入

经典方式输入是以行为导向的。输入字符存放于用户可编辑的缓冲中，直到接收到 CR (回车)或 LF (进行)。

当选择这个模式时，你通常会选择 *ICANON*, *ECHO*, and *ECHOE* options:

```
options.c_lflag |= (ICANON | ECHO | ECHOE);
```

选择 Raw

Raw 流方式输入没有预处理。当收到输入字符后，它们不经任何处理地通过。当使用 raw 流方式输入时，通常你需要去除 *ICANON*, *ECHO*, *ECHOE*, 和 *ISIG* 选项：

```
options.c_lflag &= ~(ICANON | ECHO | ECHOE | ISIG);
```

一个输入回显的提示

当发送指令到 MODEM 或其它会回显字符的计算机时，决不 使用输入回显 (*ECHO*, *ECHOE*)，否则你将造成在两个串口间回显循环。

输入项

输入模式设置项 *c_iflag* 控制处理在端口接收到的任何字符。类似 *c_cflag* 域，用位操作 OR 连接希望设置的常量，并最终存储于 *c_iflag* 。

表 7 - <i>c_iflag</i> 设置项常量	
常量	描述
INPCK	启用校验检查
IGNPAR	忽略校验错误
PARMRK	标记校验错误
ISTRIP	剥离校验位
IXON	启用软件流控制（输出）
IXOFF	启用软件流控制（输入）
IXANY	允许任何字符，再次启动字符流
IGNBRK	忽略终端条件
BRKINT	当检测到中断，发送一个 SIGINT
INLCR	将 NL 转换为 CR
IGNCR	忽略 CR
ICRNL	将 CR 转换为 NL
IUCLC	将大写字符转换为小写
IMAXBEL	输入行过长时候，回显 BEL

设置输入校验项

当你在 *c_cflag* 设置项中启用了校验 (*PARENB*)，你就应该做输入校验检查。输入校验检查的相关常量是 *INPCK*, *IGNPAR*, *PARMRK* , 和 *ISTRIP*。一般你会选择 *INPCK* 和 *ISTRIP* 来检查和剥离校验位：

```
options.c_iflag |= (INPCK | ISTRIP);
```

IGNPAR 是个稍微有点危险的选项，告诉串行驱动忽略校验错误并传送输入的数据，只当没有错误发生。在测试通讯链路质量时会有用，但一般没有在实际活动中使用的理由。

PARMRK 使用特殊字符将输入流中校验错误标记'marked'。如果 *IGNPAR* 作用，在每个有校验错误的字符前插入一个 NUL 字符 (000 八进制)。除此之外，伴随错误字符发送 DEL (177 八进制) 和 NUL 字符。

设置软件流控制

软件流控制通过设置 *IXON*, *IXOFF* 和 *IXANY* 常量启用：

```
options.c_iflag |= (IXON | IXOFF | IXANY);
```

关闭软件流控制，可以如下设置掩码：

```
options.c_iflag &= ~(IXON | IXOFF | IXANY);
```


XON (开始传送数据) 和 XOFF (停止数据传送) 的字符在 `c_cc` 数组中定义，后文将说明。

输出项

`c_oflag` 设置项包含输出过滤项。类似输入模式，你能选择预处理或 `raw` 流方式数据输出。

表 8 - <code>c_oflag</code> 设置项常量	
常量	描述
OPOST	输出后处理(不设置，即 <code>raw</code> 流输出)
OLCUC	将小写字符转换为大写
ONLCR	将 <code>NL</code> 转换为 <code>CR-NL</code>
OCRNL	将 <code>CR</code> 转换为 <code>NL</code>
NOCR	在 0 号列，没有 <code>CR</code> 输出
ONLRET	<code>NL</code> 表现为 <code>CR</code> 功能
OFILL	用字符填充延时
OFDEL	用作填充的字符是 <code>DEL</code>
NLDLY	定义行间延时，所使用的掩码
NL0	<code>NL</code> 没有延时
NL1	换新行后，延时 100ms
CRDLY	回车返回最左侧的列的延时，所使用的时间掩码
CR0	<code>CR</code> 没有延时
CR1	依据当前的列位置，在 <code>CR</code> 后所要延时
CR2	在 <code>CR</code> 后，延时 100ms
CR3	在 <code>CR</code> 后，延时 150ms
TABDLY	在 <code>TAB</code> 后延时，所使用的掩码
TAB0	<code>TAB</code> 后没有延时
TAB1	依据当前 <code>TAB</code> 所在列位置，所设置的延时
TAB2	在 <code>TAB</code> 后，延时 100ms
TAB3	将 <code>TAB</code> 字符扩展为空格
BSDLY	在 <code>BS</code> 后延时，所使用的掩码
BS0	<code>BS</code> 后没有延时
BS1	在 <code>BS</code> 后延时 50ms
VTDLY	在 <code>VT</code> 后延时，所使用的掩码
VT0	在 <code>VT</code> 后没有延时
VT1	在 <code>VT</code> 后，延时 2 秒
FFDLY	在 <code>FF</code> 后延时，所使用的掩码
FF0	在 <code>FF</code> 没有延时
FF1	在 <code>FF</code> 后，延时 2 秒

选择输出处理

输出处理是通过选择 `c_oflag` 设置项中的 `OPOST` :

```
options.c_oflag |= OPOST;
```

在所有不同的选项中，你可能只会用到 `ONLCR` 项，该项将换行替换为 `CR-LF` 对。余下的输出选项主要都是出于历史原因，可以追溯到打印机和终端的串行数据流还不能保持一致的时候！

选择 **Raw** 流方式输出

Raw 流方式输出是通过重置 `c_oflag` 成员中的 `OPOST` 项来实现的。

```
options.c_oflag &= ~OPOST;
```

当 `OPOST` 选项被设为无效，所有其它在 `c_oflag` 中的选项都被忽略。

控制字符

`c_cc` 字符数组包含了控制字符和超时参数。数组的每个元素都有常量定义。

表 9 - <code>c_cc</code> 设置项的控制字符		
常量	描述	KEY
VINTR	中断	CTRL-C
VQUIT	退出	CTRL-Z
VERASE	擦除	Backspace (BS)
VKILL	删除行	CTRL-U
VEOF	文件结束	CTRL-D
VEOL	行结束	Carriage return (CR)
VEOL2	第二个行结束	Line feed (LF)
VMIN	读取的最小字符数	-
VSTART	流开始	CTRL-Q (XON)
VSTOP	流结束	CTRL-S (XOFF)
VTIME	等待数据的时间（10 秒为单位）	-

设置软件流控制字符

`c_cc` 数组的 `VSTART` 和 `VSTOP` 元素包含用于软件流控制的字符。通常它们设定为 `DC1` (`021` 八进制) 和 `DC3` (`023` 八进制)，代表 **ASCII** 标准的 `XON` 和 `XOFF` 字符。

设置读取超时

UNIX 串行接口驱动有能力指定字符和包的超时。`c_cc` 数组的两个元素用于该目的：`VMIN` 和 `VTIME`。对于通过 `open` 或 `fcntl` 的文件，在经典输入模式下，或当 `NDELAY` 项被设置，忽略超时设置。

VMIN 指明了最小的读取字符数。如果它被设置为 0, *VTIME* 的值用于指明等待读取每个字符的时间。注意这不是意味着 *read* 调用读取 N 个字节需要等待 N 个字符进入。而是, 超时将仅对第一个字符作用, 并且 *read* 调用将立即返回字符的个数 (不超过你要求的数目)。

如果 *VMIN* 非 0, *VTIME* 指明了等待第一个字符到来的时间。如果在指定时间内读到一个字符, 任何读操作将阻塞 (等待)直到所有 *VMIN* 规定的字符数读到。那就是说, 一旦第一个字符读到, 串行接口驱动期望接收一整个包的字符 (*VMIN* 规定字符数)。如果在规定时间内没有字符被读取, *read* 调用返回 0。该方法允许你告诉串行驱动, 你准确需要 N 个字节, 以及 *read* 调用将只返回 0 或 N 字节。然而, 超时只对第一个字符作用, 所以如果由于某些原因驱动丢失了 N 个字节长的包中的某个字符, *read* 调用将永远阻塞等待后续字符的输入。

VTIME 指明等待输入的时间, 以十分之一秒为单位。如果 *VTIME* 设置为 0 (缺省), 读操作将不确定地阻塞(等待), 除非在用 *open* or *fcntl* 时, 在端口上设置 *NDELAY* 项。

第三章，MODEM 通讯

本章包括了拨号电话调制/解调器(MODEM)的通讯基础知识。使用 MODEM 的例子采用的是事实上的行业标准 AT 指令。

什么是 MODEM?

MODEM 是一种可将串行数据转换为频率信号，从而可以在模拟数据线路（例如，电话线或 TV 电缆接线）上传输的设备。一个标准电话 MODEM 转换串行数据 为语音，从而能够在电话线上传送，由于速度和复杂的转换机制，如果你试图听这些话，会发现其听起来象高声尖叫。

今天的电话 MODEM 能够在电话线上以每秒 53,000 比特的速率传输数据，或者说是 53kbps。此外，多数 MODEM 使用数据压缩技术，从而可将某些类型的比特速率提高到 100kbps 以上。

与 MODEM 通讯

用 MODEM 通讯的第一步，是打开端口并设置为行方式输入，就象[列表 3](#) 中所示。

列表 3 - 为 raw 流式输入设置端口

```
int          fd;
struct termios options;

/* 打开端口 */
fd = open("/dev/ttyS0", O_RDWR | O_NOCTTY | O_NDELAY);
fcntl(fd, F_SETFL, 0);

/* 获得当前选项 */
tcgetattr(fd, &options);

/* 设置 raw 流方式输入，超时为 1 秒 */
options.c_cflag |= (CLOCAL | CREAD);
options.c_lflag  &= ~(ICANON | ECHO | ECHOE | ISIG);
options.c_oflag  &= ~OPOST;
options.c_cc[VMIN] = 0;
options.c_cc[VTIME] = 10;

/* 写入设置项 */
tcsetattr(fd, TCSANOW, &options);
```

下一步是建立与 MODEM 的通讯。最好的办法，是发送"AT"指令到 MODEM。这还允许智能 MODEM 检测你使用的波特率。当 MODEM 正确连接和加电，它将返回"OK"。

列表 4 - 初始化 MODEM.

```
int          /* 0 - 0 = MODEM ok, -1 = MODEM bad */
init_modem(int fd) /* I - 串行接口文件 */
{
```

```

char buffer[255]; /* 输入缓冲 */
char *bufptr;     /* 当前字符在缓冲中的位置 */
int nbytes;       /* 读入的字节数 */
int tries;        /* 尝试的次数 */

for (tries = 0; tries < 3; tries++)
{
    /* 发送一个 CR 结尾的 AT 命令 */
    if (write(fd, "AT\r", 3) < 3)
        continue;

    /* 读取字符到字符串缓冲，直到遇到 CR 或 NL 作为结尾 */
    bufptr = buffer;
    while ((nbytes = read(fd, bufptr, buffer + sizeof(buffer) - bufptr - 1)) > 0)
    {
        bufptr += nbytes;
        if (bufptr[-1] == '\n' || bufptr[-1] == '\r')
            break;
    }

    /* nul 作为字符串的几位，并检查是否受到一个 OK 作为结尾相应 */
    *bufptr = '\0';

    if (strncmp(buffer, "OK", 2) == 0)
        return (0);
}

return (-1);
}

```

标准 MODEM 指令

多数 MODEM 支持"AT"指令集，这么称是因为每个指令都以"AT"开头。每个指令都以"AT"字符开头，跟以特殊指令和回车结尾(CR, 015 octal)。处理指令之后，MODEM 将根据指令返回若干文本信息之一。

ATD – 拨一个号码

ATD 指令拨一个指定号码。除数字外，你还能添加少量内容，包括指明是语音("T")或脉冲("P")拨号，等待一秒(",")和等待拨号音("W")：

```

ATDT 555-1212
ATDT 18008008008W1234,1,1234
ATD T555-1212WP1234

```

MODEM 将返回以下信息之一：

```

NO DIALTONE
BUSY
NO CARRIER
CONNECT
CONNECT baud

```

ATH – 挂起

ATH 指令会 MODEM 挂起。因为 MODEM 必须是处于"命令"模式，你通常是不会在普通通话中这么做的。

如果 DTR 信号下降，多数的 MODEM 会挂起；你可以通过将波特率设置为 0，并持续至少一秒来实现。降低 DTR 信号，同时也将 MODEM 返回命令模式。

在成功挂起后，MODEM 将返回 "NO CARRIER" 没有载波。如果 MODEM 依然连接着，将传回 "CONNECT" 或 "CONNECT BAUD"。

ATZ – 重置 MODEM

ATZ 指令将重置 MODEM，MODEM 将返回字符"OK"。

MODEM 常见通讯问题

首先和也是最重要的，*不要忘记关闭输入回显*。输入回显将导致 MODEM 和计算机间形成反馈回路。

其次，在传输指令到 MODEM 时，必须以回车(CR)结尾，而不是新行(NL)。C 对于 CR 的字符常量是 "\r"。

最后，当操作 MODEM 时，请确认你使用的波特率是 MODEM 所支持的。你必须注意，尽管许多 MODEM 能够自动检测波特率，而一些有限制（通常在老式 MODEM 上是 19.2kbps）。

第四章，高级串行编程

本章包括了使用 *ioctl(2)* 和 *select(2)* 系统调用，实现的高级串行编程技术。

串行端口 IOCTLS

在[第二章，串行端口设置](#)中，我们使用 *tcgetattr* 和 *tcsetattr* 函数来设置串行端口。在 UNIX 下这些函数使用 *ioctl(2)* 系统调用来实现其功能。

ioctl 系统调用使用三个参数：

```
int ioctl(int fd, int request, ...);
```

fd 参数指明串行端口文件描述符。*request* 参数是一个常量定义于<termios.h> 头文件中，通常是列在[表 10](#) 中的某个。

表 10 – 作用于串行端口的 IOCTL 的请求常量		
请求	描述	POSIX 功能
TCGETS	获得当前的串行端口设置	tcgetattr
TCSETS	立即设置串行端口	tcsetattr(fd, TCSANOW, &options)
TCSETSF	在刷新输入和输出缓冲后，设置串行端口	tcsetattr(fd, TCSAFLUSH, &options)
TCSETSW	允许在输入和输出缓冲排空后，设置串行端口	tcsetattr(fd, TCSADRAIN, &options)
TCSBRK	在指定时间内，发送一个中断	tcsendbreak, tcdrain
TCXONC	控制软件流控制	tcflow
TCFLSH	刷新输入和输出缓冲	tcflush
TIOCMGET	返回 "MODEM" 位的状态.	None
TIOCMSET	设置 "MODEM" 位的状态	None
FIONREAD	返回输入缓冲中的字节数目	None

获得控制信号

TIOCMGET *ioctl* 获得当前 "MODEM" 状态位，包含所有列在[表 11](#) 中的 RS-232 信号线，除了 **RXD** 和 **TXD**。

要获得状态位，用一个整数指针来调用 *ioctl* 并保存数据位，如[列表 5](#) 。

列表 5 – 获得 MODEM 状态位

```
#include <unistd.h>
#include <termios.h>
```

```
int fd;
int status;

ioctl(fd, TIOCMGET, &status);
```

表 11 – 控制信号常量	
常量	描述
TIOCM_LE	DSR (data set ready/line enable)
TIOCM_DTR	DTR (data terminal ready)
TIOCM_RTS	RTS (request to send)
TIOCM_ST	第二个 TXD (transmit)
TIOCM_SR	第二个 RXD (receive)
TIOCM_CTS	CTS (clear to send)
TIOCM_CAR	DCD (data carrier detect)
TIOCM_CD	Synonym for TIOCM_CAR
TIOCM_RNG	RNG (ring)
TIOCM_RI	Synonym for TIOCM_RNG
TIOCM_DSR	DSR (data set ready)

设定控制信号

TIOCMSET *ioctl* 设置如上 "MODEM" 状态。要降低 DTR 信号，你可以使用[列表 6](#) 中的代码。

列表 6 - 使用 TIOCMSET *ioctl* 降低 DTR 信号。

```
#include <unistd.h>
#include <termios.h>

int fd;
int status;

ioctl(fd, TIOCMGET, &status);

status &= ~TIOCM_DTR;

ioctl(fd, TIOCMSET, &status);
```

能够设定的位依赖于操作系统，驱动和使用的模式。请查询你的操作系统文档以获得详细信息。

获得有效的比特字节个数

FIONREAD *ioctl* 获得串行端口输入缓冲的字节个数。你调用 TIOCMGET 的同时，用一个整数指针获得字节个数，例如[列表 7](#)。

列表 7 - 获得在输入缓冲中的字节个数

```
#include <unistd.h>
```



```
#include <termios.h>

int fd;
int bytes;

ioctl(fd, FIONREAD, &bytes);
```

这在用串行端口获得数据时很有用，使你的程序能够在尝试读操作前，先确定字节个数。

从串行端口选择输入

虽然，简单的应用程序能够获得或等待串行端口的数据，多数应用程序并不简单，可能需从多个来源获得输入。

UNIX 通过 *select(2)* 系统调用提供这一能力。系统调用允许你的程序在一或多个文件描述符上，检查输入，输出或错误条件。文件描述符能够指向串行端口，普通文件，其它设备，管道或 *socket* 端口。你能使用 *select* 系统调用非常灵活地检查挂起的输入，等待输入结束，或在指定时间后超时，。

多数 GUI 工具提供了一个界面访问 *select*；我们将在本章稍后讨论 X Intrinsics ("Xt")库。

SELECT 系统调用

select 系统调用能够接受 5 个参数：

```
int select(int max_fd, fd_set *input, fd_set *output, fd_set *error,
           struct timeval *timeout);
```

max_fd 参数指定了，在 *input*, *output*, 和 *error* 三个文件描述符集合中最大值。*input*, *output*, 和 *error* 参数指明了一个文件描述的集合，分别用于操作未处理输入，输出或错误条件的；指定 NULL 来禁止监测相关条件。这些设置用以下三个宏初始化：

```
FD_ZERO(&fd_set);
FD_SET(fd, &fd_set);
FD_CLR(fd, &fd_set);
```

FD_ZERO 宏清除所有设定。*FD_SET* 和 *FD_CLR* 宏从集合中，添加和删除相关的文件描述符。

超时 *timeout* 参数指明了超时设置，由秒(*timeout.tv_sec*) 和毫秒(*timeout.tv_usec*) 组成。设置秒和毫秒为零，可从一或多个文件描述符来源获得数据。要无限等待，指定超时指针为 NULL。

select 系统调用返回未处理的文件描述符个数，或-1 代表错误。

使用 SELECT 系统调用

假定，我们从一个串行端口和一个套接字读取数据。我们从每个文件描述符检查输入，并希望如果在 10 秒内没有输入时提醒用户。要实现这些，我们需要使用 *select* 系统调用，如[列表 8](#)。

列表 8 - 使用 *SELECT* 处理一个以上的输入来源。

```
#include <unistd.h>
#include <sys/types.h>
```

```

#include <sys/time.h>
#include <sys/select.h>

int      n;
int      socket;
int      fd;
int      max_fd;
fd_set   input;
struct timeval timeout;

/* 初始化输入 */
FD_ZERO(&input);
FD_SET(fd, &input);
FD_SET(sock, &input);

max_fd = (sock > fd ? sock : fd) + 1;

/* 初始化超时结构设置 */
timeout.tv_sec = 10;
timeout.tv_usec = 0;

/* Do the select */
n = select(max_fd, &input, NULL, NULL, &timeout);

/* 看是否出错 */
if (n < 0)
    perror("select failed");
else if (n == 0)
    puts("TIMEOUT");
else
{
    /* 有输入 */
    if (FD_ISSET(fd, &input))
        process_fd();
    if (FD_ISSET(sock, &input))
        process_socket();
}

```

你会注意到，我们首先检查 *select* 系统调用的返回值。0 值和 -1 值分别导致警告和超时错误。大于 0 的值表明有一或多个未处理的值。

要确定哪个文件描述符有没处理的输入，我们使用 *FD_ISSET* 宏来测试每个文件描述符的输入集合。如果文件描述符设定了，且设定的条件出现(这里就是有未处理的输入)，我们就需要做点什么。

配合 X INTRINSICS 库，使用 SELECT 系统调用

X Intrinsics 库提供了一个 *select* 系统调用的接口，通过 *XtAppAddInput(3x)* 和 *XtAppRemoveInput(3x)* 函数：

```

int XtAppAddInput(XtAppContext context, int fd, int mask,
                  XtInputProc proc, XtPointer data);
void XtAppRemoveInput(XtAppContext context, int input);

```

select 系统调用是在内部使用，用于实现来自 X server 的超时、工作过程和输入检查。这些函数能够与任何基于 Xt 的工具一同使用，包括 Xaw, Lesstif, and Motif。

XtAppAddInput 的 *proc* 参数指明了，当在文件描述符上设定的条件出现时(例如，有输入)，调用的函数。类似前面的例子，你可以用 *process_fd* 或 *process_socket* 函数。

因为 Xt 限制了你访问 *select* 系统调用，你需要用其它方法实现超时设置，例如通过 *XtAppAddTimeout(3x)*。

附录 A,引脚定义

这个附录提供了你经常会发现的许多串口引脚定义。

RS-232 引脚

RS-232 有三种类型 (A, B, C) 并使用 25-pin D 形接头：

图 2 - RS-232 连接头

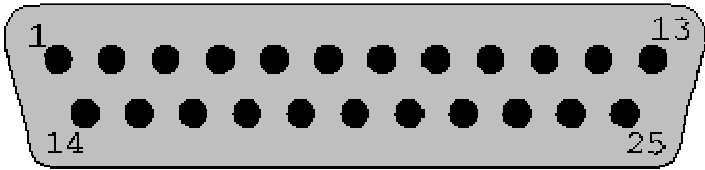


表 12 - RS-232 信号

PIN	描述	PIN	描述
1	Earth Ground	14	第二个 TXD
2	TXD - Transmitted Data	15	Transmit Clock
3	RXD - Received Data	16	第二个 RXD
4	RTS - Request To Send	17	Receiver Clock
5	CTS - Clear To Send	18	未定义
6	DSR - Data Set Ready	19	第二个 RTS
7	GND - Logic Ground	20	DTR - Data Terminal Ready
8	DCD - Data Carrier Detect	21	Signal Quality Detect
9	保留	22	Ring Detect
10	保留	23	Data Rate Select
11	未定义	24	Transmit Clock
12	第二个 DCD	25	未定义
13	第二个 CTS		

RS-422 引脚

RS-422 也使用 25-pin D 形街头，但信号不同：

图 3 - RS-422 连接头

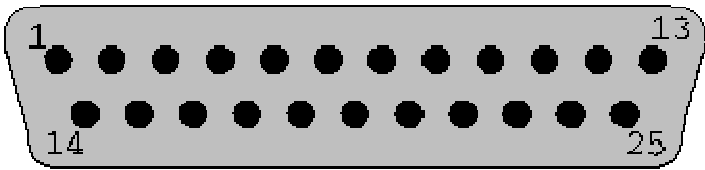


表 13 - RS-422 信号

PIN	描述	PIN	描述
1	Earth Ground	14	TXD+
2	TXD- - Transmitted Data	15	Transmit Clock-
3	RXD- - Received Data	16	RXD+
4	RTS- - Request To Send	17	Receiver Clock-
5	CTS- - Clear To Send	18	未定义
6	DSR - Data Set Ready	19	RTS+
7	GND - Logic Ground	20	DTR- - Data Terminal Ready
8	DCD- - Data Carrier Detect	21	Signal Quality Detect
9	保留	22	未定义
10	保留	23	DTR+
11	未定义	24	Transmit Clock+
12	DCD+	25	Receiver Clock+
13	CTS+		

RS-574 (IBM PC/AT) 引脚

RS-574 街头主要是由 PC 厂商使用，是 9-pin D 形接头：

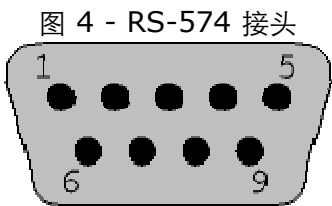


表 14 - RS-574 (IBM PC/AT) 信号			
PIN	描述	PIN	描述
1	DCD - Data Carrier Detect	6	Data Set Ready
2	RXD - Received Data	7	RTS - Request To Send
3	TXD - Transmitted Data	8	CTS - Clear To Send
4	DTR - Data Terminal Ready	9	Ring Detect
5	GND - Logic Ground		

SGI 引脚

老式 SGI 设备也使用 9-pin D 形接头，不用于 RS-574，SGI 引脚很接近 RS-232 标准：

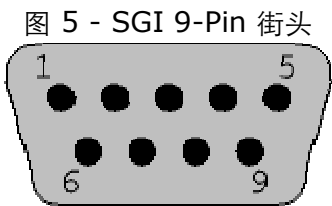


表 15 - SGI 9-Pin D 接头型号

PIN	描述	PIN	描述
1	Earth Ground	6	DSR - Data Set Ready
2	TXD - Transmitted Data	7	GND - Logic Ground
3	RXD - Received Data	8	DCD - Data Carrier Detect
4	RTS - Request To Send	9	DTR - Data Terminal Ready
5	CTS - Clear To Send		

The SGI Indigo, Indigo2, and Indy workstations use the Apple 8-pin MiniDIN connector for their serial ports:

Figure 6 - SGI 8-Pin Connector

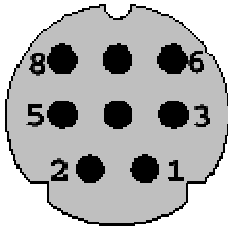


Table 16 - SGI 8-Pin MiniDIN Signals			
PIN	描述	PIN	描述
1	DTR - Data Terminal Ready	5	RXD - Received Data
2	CTS - Clear To Send	6	RTS - Request To Send
3	TXD - Transmitted Data	7	DCD - Data Carrier Detect
4	GND - Logic Ground	8	GND - Logic Ground

附录 B，ASCII 控制码

本章列出了 ASCII 控制代码和他们的名字。

控制代码

以下的 ASCII 字符被用于控制目的：

表 17 - ASCII 控制码				
NAME	BINARY	OCTAL	DECIMAL	HEXADECIMAL
NUL	00000000	000	0	00
SOH	00000001	001	1	01
STX	00000010	002	2	02
ETX	00000011	003	3	03
EOT	00000100	004	4	04
ENQ	00000101	005	5	05
ACK	00000110	006	6	06
BEL	00000111	007	7	07
BS	00001000	010	8	08
HT	00001001	011	9	09
NL	00001010	012	10	0A
VT	00001011	013	11	0B
NP, FF	00001100	014	12	0C
NAME	BINARY	OCTAL	DECIMAL	HEXADECIMAL
CR	00001101	015	13	0D
SO	00001110	016	14	0E
SI	00001111	017	15	0F
DLE	00010000	020	16	10
XON, DC1	00010001	021	17	11
DC2	00010010	022	18	12
XOFF, DC3	00010011	023	19	13
DC4	00010100	024	20	14
NAK	00010101	025	21	15
SYN	00010110	026	22	16
ETB	00010111	027	23	17
CAN	00011000	030	24	18
EM	00011001	031	25	19
SUB	00011010	032	26	1A
ESC	00011011	033	27	1B

FS	00011100	034	28	1C
GS	00011101	035	29	1D
RS	00011110	036	30	1E
US	00011111	037	31	1F

APPENDIX C, GNU FREE DOCUMENTATION LICENSE

Version 1.2, November 2002

Copyright (C) 2000,2001,2002 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released

under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you

distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- **A.** Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- **B.** List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- **C.** State on the Title page the name of the publisher of the Modified Version, as the publisher.
- **D.** Preserve all the copyright notices of the Document.

- **E.** Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- **F.** Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- **G.** Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- **H.** Include an unaltered copy of this License.
- **I.** Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- **J.** Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- **K.** For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- **L.** Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- **M.** Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- **N.** Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- **O.** Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements."

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers.

In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

HOW TO USE THIS LICENSE FOR YOUR DOCUMENTS

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright (c) YEAR YOUR NAME.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.2
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover
Texts. A copy of the license is included in the section entitled "GNU
Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

with the Invariant Sections being LIST THEIR TITLES, with the
Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

附录 D，修改历史

This appendix lists the changes that have been made in this edition.

EDITION 5, REVISION 6

The following changes were made for the 6th revision:

- The select() example did not correctly use the FD macros.
- The title page and this appendix were not properly updated.

EDITION 5, REVISION 5

The following changes were made for the 5th revision:

- The select() documentation did not correctly describe the FD macros.
- Appendix C was missing the "how to use" part of the GNU FDL.

EDITION 5, REVISION 4

The following changes were made for the 4th revision:

- Changed the 描述 of the read() system call semantics in chapter 1.
- Added 描述 s for VSTART and VSTOP to chapter 2.

EDITION 5, REVISION 3

The following changes were made for the 3rd revision:

- Now use the GNU Free Documentation License for the guide.
- Changed the examples to use the Linux serial port filenames.
- Put the infrastructure in place to allow for easier translations of the guide.
- The guide text is now fully justified.