**termios结构体内容：**

**成员 描述**

-------------------------------------------

c\_cflag 控制模式标志

c\_lflag 本地模式标志

c\_iflag 输入模式标志

c\_oflag 输出模式标志

c\_line line discipline

c\_cc[NCCS] 控制字符

c\_ispeed 输入波特率

c\_ospeed 输出波特率

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

在termios结构中的四个标志控制了输入输出的四个不同部分。输入模式标志c\_iflag决定如何解释和处理接收的字符。输出模式标志c\_oflag决定如何解释和处理发送到tty设备的字符。控制模式标志决定设备的一系列协议特征，这一标志只对物理设备有效。本地模式标志c\_lflag决定字符在输出前如何收集和处理。

在串口传输中，用波特率来表示传输的速度，1波特表示在1秒钟内可以传输1个码元。波特率设置可以使用 cfsetispeed(&new\_termios,B19200)和 cfsetospeed(&new\_termios,B19200)这两个函数来完成，默认的波特率为9600baud。 cfsetispeed()函数用来设置输入的波特率，cfsetospeed()函数用来设置输出的波特率。B19200是termios.h头文件里 定义的一个宏，表示19200的波特率。

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

CLOCAL和CREAD是c\_cflag成员中与速率相关的标志，在串口编程中，这两个标志一定要有效，以确保程序在突发的作业控制或挂起时，不会成为端口的占有都，同时串口的接收驱动会自动读入数据。设置方法如下：

termios\_new.c\_cflag |= CLOCAL; //保证程序不会成为端的占有者

termios\_new.c\_cflag |= CREAD; //使端口能读取输入的数据

设置串口属性不能直接赋值，要通过对termios不同成员进行"与"和"或"操作来实现。在termios.h文件，定义了各种常量，如上面介绍 的CLOCAL，CREAD。这些常量的值是掩码，通过把这些常量与termios结构成员进行逻辑操作就可实现串口属性的设置。在编程时用**"|="来启 用属性，用"&=~"来取消**属性。

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

c\_iflag 输入标志说明

BRKINT和IGNBRK

如果设置了IGNBRK，中断条件被忽略。如果没有设置IGNBRK而设置了BRKINT，中断条件清空输入输出队列中所有的数据并且向tty的前台进程组中所有进程发送一个SIGINT信号。如果这两个都没有设置，中断条件会被看作一个0字符。这时，如果设置了PARMRK，当检测到一个帧误差时 将会向应用程序发送三个字节'/377''/0''/0'，而不是只发送一个'/0'。

PARMRK和IGNPAR

如果设定了IGNPAR，则忽略接收到的数据的奇偶检验错误或帧错误（除了前面提到的中断条件）。如果没有设置IGNPAR而设置了PARMRK， 当接收到的字节存在奇偶检验错误或帧错误的时候。将向应用程序发送一个三字节的'/377''/0''/n'错误报告。其中n表示所接收到的字节。如果两 者都没有设置，除了接收到的字节存在奇偶检验错误或帧误差之外的中止条件都会向应用程序发送一个单字节（'/0'）的报告。

INPCK

如果设置，则进行奇偶校验。如果不进行奇偶检验，PARMRK和IGNPAR将对存在的奇偶校验错误不产生任何的影响。

ISTRIP

如果设置，所接收到的所有字节的高位将会被去除，保证它们是一个7位的字符。

INLCR

如果设置，所接收到的换行字符（'/n'）将会被转换成回车符（'/r'）。

IGNCR

如果设置，则会忽略所有接收的回车符（'/r'）。

ICRNL

如果设置，但IGNCR没有设置，接收到的回车符向应用程序发送时会变换成换行符。

IUCLC

如果IUCLC和IEXTEN都设置，接收到的所有大写字母发送给应程序时都被转换成小写字母。POSIX中没有定义该标记。

IXOFF

如果设置，为避免tty设备的输入缓冲区溢出，tty设备可以向终端发送停止符^S和开始符^Q，要求终端停止或重新开始向计算机发送数据。通过停 止符和开始符来控制数据流的方式叫软件流控制，软件流控制方式较少用，我们主要还是用硬件流控制方式。硬件流控制在c\_cflag标志中设置。

IXON

如果设置，接收到^S后会停止向这个tty设备输出，接收到^Q后会恢复输出。

IXANY

如果设置，则接到任何字符都会重新开始输出，而不仅仅是^Q字符。

IMAXBEL

如果设置，当输入缓冲区空间满时，再接收到的任何字符就会发出警报符'/a'。POSIX中没有定义该标志。

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

c\_oflag 输出标志说明

OPOST是POSIX定义的唯一一个标志，只有设置了该标志后，其它非POSIX的输出标记才会生效。

OPOST

开启该标记，后面的输出标记才会生效。否则，不会对输出数据进行处理。

OLCUC

如果设置，大写字母被转换成小写字母输出。

ONLCR

如果设置，在发送换行符（'/n'）前先发送回车符（'/r'）。

ONOCR

如果设置，当current column为0时，回车符不会被发送也不会被处理。

OCRNL

如果设置，回车符会被转换成换行符。另外，如果设置了ONLRET，则current column会被设为0.

ONLRET

如果设置，当一个换行符或回车符被发送的时候，current column会被设置为0。

OXTABS

如果设置，制表符会被转换成空格符

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

c\_cflag 控制模式标志说明

CLOCAL

如果设置，modem的控制线将会被忽略。如果没有设置，则open()函数会阻塞直到载波检测线宣告modem处于摘机状态为止。

CREAD

只有设置了才能接收字符，该标记是一定要设置的。

CSIZE

设置传输字符的位数。CS5表示每个字符5位，CS6表示每个字符6位，CS7表示每个字符7位，CS8表示每个字符8位。

CSTOPB

设置停止位的位数，如果设置，则会在每帧后产生两个停止位，如果没有设置，则产生一个停止位。一般都是使用一位停止位。需要两位停止位的设备已过时了。

HUPCL

如果设置，当设备最后打开的文件描述符关闭时，串口上的DTR和RTS线会减弱信号，通知Modem挂断。也就是说，当一个用户通过Modem拔号 登录系统，然后注销，这时Modem会自动挂断。

PARENB和PARODD

如果设置PARENB，会产生一个奇偶检验位。如果没有设置PARODD，则产生偶校验位，如果设置了PARODD，则产生奇校验位。如果没有设置 PARENB，则PARODD的设置会被忽略。

CRTSCTS

使用硬件流控制。在高速（19200bps或更高）传输时，使用软件流控制会使效率降低，这个时候必须使用硬件流控制。

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

c\_cc[] 控制字符说明

只有在本地模式标志c\_lflag中设置了IEXITEN时，POSIX没有定义的控制字符才能在Linux中使用。每个控制字符都对应一个按键组 合（^C、^H等），但VMIN和VTIME这两个控制字符除外，它们不对应控制符。这两个控制字符只在原始模式下才有效。

c\_cc[VINTR]

默认对应的控制符是^C，作用是清空输入和输出队列的数据并且向tty设备的前台进程组中的每一个程序发送一个SIGINT信号，对SIGINT信 号没有定义处理程序的进程会马上退出。

c\_cc[VQUIT]

默认对应的控制符是^/，作用是清空输入和输出队列的数据并向tty设备的前台进程组中的每一个程序发送一个SIGQUIT信号，对SIGQUIT 信号没有定义处理程序的进程会马上退出。

c\_cc[verase]

默认对应的控制符是^H或^?，作用是在标准模式下，删除本行前一个字符，该字符在原始模式下没有作用。

c\_cc[VKILL]

默认对应的控制符是^U，在标准模式下，删除整行字符，该字符在原始模式下没有作用。

c\_cc[VEOF]

默认对应的控制符是^D，在标准模式下，使用read()返回0，标志一个文件结束。

c\_cc[VSTOP]

默认对应的控制字符是^S，作用是使用tty设备暂停输出直到接收到VSTART控制字符。或者，如果设备了IXANY，则等收到任何字符就开始输 出。

c\_cc[VSTART]

默认对应的控制字符是^Q，作用是重新开始被暂停的tty设备的输出。

c\_cc[VSUSP]

默认对应的控制字符是^Z，使当前的前台进程接收到一个SIGTSTP信号。

c\_cc[VEOL]和c\_cc[VEOL2]

在标准模式下，这两个下标在行的末尾加上一个换行符（'/n'），标志一个行的结束，从而使用缓冲区中的数据被发送，并开始新的一行。POSIX中 没有定义VEOL2。

c\_cc[VREPRINT]

默认对应的控制符是^R，在标准模式下，如果设置了本地模式标志ECHO，使用VERPRINT对应的控制符和换行符在本地显示，并且重新打印当前 缓冲区中的字符。POSIX中没有定义VERPRINT。

c\_cc[VWERASE]

默认对应的控制字符是^W，在标准模式下，删除缓冲区末端的所有空格符，然后删除与之相邻的非空格符，从而起到在一行中删除前一个单词的效果。 POSIX中没有定义VWERASE。

c\_cc[VLNEXT]

默认对应的控制符是^V，作用是让下一个字符原封不动地进入缓冲区。如果要让^V字符进入缓冲区，需要按两下^V。POSIX中没有定义 VLNEXT。

要禁用某个控制字符，只需把它设置为\_POSIX\_VDISABLE即可。但该常量只在Linux中有效，所以如果程序要考虑移植性的问题，请不要 使用该常量。

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

c\_lflag 本地模式标志说明

ICANON

如果设置，则启动标准模式，如果没有设置，则启动原始模式。

ECHO

如果设置，则启动本地回显。如果没有设置，则除了ECHONL之外，其他以ECHO开头的标记都会失效。

ECHOCTL

如果设置，则以^C的形式打印控制字符，如：按Ctrl+C显示^C，按Ctrl+？显示^?。

ECHOE

如果在标准模式下设定了ECHOE标志，则当收到一个ERASE控制符时将删除前一个显示字符。

ECHOK和ECHOKE

在标准模式下，当接收到一个KILL控制符，则在缓冲区中删除当前行。如果ECHOK、ECHOKE和ECHOE都没有设置，则用ECHOCTL表 示的KILL字符（^U）将会在输出终端上显示，表示当前行已经被删除。

如果已经设置了ECHOE和ECHOK，但没有设置ECHOKE，将会在输出终端显示ECHOCTL表示的KILL字符，紧接着是换行，如果设置了 OPOST，将会通过OPOST处理程序进行适当的处理。

如果ECHOK、ECHOKE和ECHOE都有设置，则会删除当前行。

在POSIX中没有定义ECHOKE标记，在没有定义ECHOKE标记的系统中，设置ECHOK则表示同时设置了ECHOKE标志。

ECHONL

如果在标准模式下设置了该标志，即使没有设置ECHO标志，换行符还是会被显示出来。

ECHOPRT

如果设置，则字符会被简单地打印出来，包括各种控制字符。在POSIX中没有定义该标志。

ISIG

如果设置，与INTR、QUIT和SUSP相对应的信号SIGINT、SIGQUIT和SIGTSTP会发送到tty设备的前台进程组中的所有进 程。

NOFLSH

一般情况下，当接收到INTR或QUIT控制符的时候会清空输入输出队列，当接收到SUSP控制符时会清空输入队列。但是如果设置了NOFLUSH 标志，则所有队列都不会被清空。

TOSTOP

如果设置，则当一个非前台进程组的进程试图向它的控制终端写入数据时，信号SIGTTOU会被被发送到这个进程所在的进程组。默认情况下，这个信号 会使进程停止，就像收到SUSP控制符一样。

IEXIEN

默认已设置，我们不应修改它。在Linux中IUCLC和几个与删除字符相关的标记都要求在设置了IEXIEN才能正常工作。