

**NAME**

**cfgetispeed**, **cfsetispeed**, **cfgetospeed**, **cfsetospeed**, **cfsetspeed**, **cfmakeraw**, **cfmakesane**, **tcgetattr**, **tcsetattr** - manipulating the termios structure

**LIBRARY**

Standard C Library (libc, -lc)

**SYNOPSIS**

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

*speed\_t*

```
cfgetispeed(const struct termios *t);
```

*int*

```
cfsetispeed(struct termios *t, speed_t speed);
```

*speed\_t*

```
cfgetospeed(const struct termios *t);
```

*int*

```
cfsetospeed(struct termios *t, speed_t speed);
```

*int*

```
cfsetspeed(struct termios *t, speed_t speed);
```

*void*

```
cfmakeraw(struct termios *t);
```

*void*

```
cfmakesane(struct termios *t);
```

*int*

```
tcgetattr(int fd, struct termios *t);
```

*int*

```
tcsetattr(int fd, int action, const struct termios *t);
```

**DESCRIPTION**

The **cfmakeraw**(), **cfmakesane**(), **tcgetattr**() and **tcsetattr**() functions are provided for getting and setting the termios structure.

The **cfgetispeed()**, **cfsetispeed()**, **cfgetospeed()**, **cfsetospeed()** and **cfsetspeed()** functions are provided for getting and setting the baud rate values in the termios structure. The effects of the functions on the terminal as described below do not become effective, nor are all errors detected, until the **tcsetattr()** function is called. Certain values for baud rates set in the termios structure and passed to **tcsetattr()** have special meanings. These are discussed in the portion of the manual page that describes the **tcsetattr()** function.

## GETTING AND SETTING THE BAUD RATE

The input and output baud rates are found in the termios structure. The unsigned integer `speed_t` is typedef'd in the include file `<termios.h>`. The value of the integer corresponds directly to the baud rate being represented, however, the following symbolic values are defined.

```
#define B0      0
#define B50    50
#define B75    75
#define B110   110
#define B134   134
#define B150   150
#define B200   200
#define B300   300
#define B600   600
#define B1200  1200
#define B1800  1800
#define B2400  2400
#define B4800  4800
#define B9600  9600
#define B19200 19200
#define B38400 38400
#ifdef _POSIX_SOURCE
#define EXTA   19200
#define EXTB   38400
#endif /* _POSIX_SOURCE */
```

The **cfgetispeed()** function returns the input baud rate in the termios structure referenced by *t*.

The **cfsetispeed()** function sets the input baud rate in the termios structure referenced by *t* to *speed*.

The **cfgetospeed()** function returns the output baud rate in the termios structure referenced by *t*.

The **cfsetospeed()** function sets the output baud rate in the termios structure referenced by *t* to *speed*.

The **cfsetspeed()** function sets both the input and output baud rate in the termios structure referenced by *t* to *speed*.

Upon successful completion, the functions **cfsetispeed()**, **cfsetospeed()**, and **cfsetspeed()** return a value of 0. Otherwise, a value of -1 is returned and the global variable *errno* is set to indicate the error.

## GETTING AND SETTING THE TERMIOS STATE

This section describes the functions that are used to control the general terminal interface. Unless otherwise noted for a specific command, these functions are restricted from use by background processes. Attempts to perform these operations shall cause the process group to be sent a SIGTTOU signal. If the calling process is blocking or ignoring SIGTTOU signals, the process is allowed to perform the operation and the SIGTTOU signal is not sent.

In all the functions, although *fd* is an open file descriptor, the functions affect the underlying terminal file, not just the open file description associated with the particular file descriptor.

The **cfmakeraw()** function sets the flags stored in the termios structure to a state disabling all input and output processing, giving a "raw I/O path", while the **cfmakesane()** function sets them to a state similar to those of a newly created terminal device. It should be noted that there is no function to reverse this effect. This is because there are a variety of processing options that could be re-enabled and the correct method is for an application to snapshot the current terminal state using the function **tcgetattr()**, setting raw or sane mode with **cfmakeraw()** or **cfmakesane()** and the subsequent **tcsetattr()**, and then using another **tcsetattr()** with the saved state to revert to the previous terminal state.

The **tcgetattr()** function copies the parameters associated with the terminal referenced by *fd* in the termios structure referenced by *t*. This function is allowed from a background process, however, the terminal attributes may be subsequently changed by a foreground process.

The **tcsetattr()** function sets the parameters associated with the terminal from the termios structure referenced by *t*. The *action* argument is one of the following values, as specified in the include file *<termios.h>*.

*TCSANOW* The change occurs immediately.

*TCSADRAIN* The change occurs after all output written to *fd* has been transmitted to the terminal. This value of *action* should be used when changing parameters that affect output.

*TCSAFLUSH*

The change occurs after all output written to *fd* has been transmitted to the terminal. Additionally, any input that has been received but not read is discarded.

The *action* may be modified by *or*'ing in *TCSASOFT* which causes the values of the *c\_cflag*, *c\_ispeed*, and *c\_ospeed* fields to be ignored.

The 0 baud rate is used to terminate the connection. If 0 is specified as the output speed to the function **tcsetattr()**, modem control will no longer be asserted on the terminal, disconnecting the terminal.

If zero is specified as the input speed to the function **tcsetattr()**, the input baud rate will be set to the same value as that specified by the output baud rate.

If **tcsetattr()** is unable to make any of the requested changes, it returns -1 and sets *errno*. Otherwise, it makes all of the requested changes it can. If the specified input and output baud rates differ and are a combination that is not supported, neither baud rate is changed.

Upon successful completion, the functions **tcgetattr()** and **tcsetattr()** return a value of 0. Otherwise, they return -1 and the global variable *errno* is set to indicate the error, as follows:

[EBADF]           The *fd* argument to **tcgetattr()** or **tcsetattr()** was not a valid file descriptor.

[EINTR]           The **tcsetattr()** function was interrupted by a signal.

[EINVAL]          The *action* argument to the **tcsetattr()** function was not valid, or an attempt was made to change an attribute represented in the *termios* structure to an unsupported value.

[ENOTTY]          The file associated with the *fd* argument to **tcgetattr()** or **tcsetattr()** is not a terminal.

## SEE ALSO

`tcsendbreak(3)`, `termios(4)`

## STANDARDS

The **cfgetispeed()**, **cfsetispeed()**, **cfgetospeed()**, **cfsetospeed()**, **tcgetattr()** and **tcsetattr()** functions are expected to be compliant with the IEEE Std 1003.1-1988 ("POSIX.1") specification. The **cfmakeraw()**, **cfmakesane()** and **cfsetspeed()** functions, as well as the *TCSASOFT* option to the **tcsetattr()** function are extensions to the IEEE Std 1003.1-1988 ("POSIX.1") specification.