NAME

ttcompat – V7, 4BSD and XENIX STREAMS compatibility module

SYNOPSIS

#include <sys/stream.h>
#include <sys/stropts.h>
#include <sys/ttold.h>
#include <sys/ttcompat.h>
#include <sys/filio.h>

ioctl(*fd,* **I_PUSH,** **"ttcompat");**

DESCRIPTION

ttcompat is a STREAMS module that translates the **ioctl** calls supported by the older **Version 7, 4BSD,** and **XENIX** terminal drivers into the **ioctl** calls supported by the **termio** interface (see **termio**(7I)). All other messages pass through this module unchanged; the behavior of **read** and **write** calls is unchanged, as is the behavior of **ioctl** calls other than the ones supported by **ttcompat**.

This module can be automatically pushed onto a stream using the **autopush** mechanism when a terminal device is opened; it does not have to be explicitly pushed onto a stream. This module requires that the **termios** interface be supported by the modules and the application can push the driver downstream. The **TCGETS, TCSETS,** and **TCSETSF ioctl** calls must be supported. If any information set or fetched by those **ioctl** calls is not supported by the modules and driver downstream, some of the **V7/4BSD/XENIX** functions may not be supported. For example, if the **CBAUD** bits in the **c_cflag** field are not supported, the functions provided by the **sg_ispeed** and **sg_ospeed** fields of the **sgttyb** structure (see below) will not be supported. If the **TCFLSH ioctl** is not supported, the function provided by the **TIOCFLUSH ioctl** will not be supported. If the **TCXONC ioctl** is not supported, the functions provided by the **TIOCSTOP** and **TIOCSTART ioctl** calls will not be supported. If the **TIOCMBIS** and **TIOCMBIC ioctl** calls are not supported, the functions provided by the **TIOCSDTR** and **TIOCCDTR ioctl** calls will not be supported.

The basic **ioctl** calls use the **sgttyb** structure defined by **<sys/ttold.h>**:

```
struct   sgttyb   {
        char          sg_ispeed;
        char          sg_ospeed;
        char          sg_erase;
        char          sg_kill;
        int           sg_flags;
};
```

The **sg_ispeed** and **sg_ospeed** fields describe the input and output speeds of the device, and reflect the values in the **c_cflag** field of the **termios** structure. The **sg_erase** and **sg_kill** fields of the argument structure specify the erase and kill characters respectively, and reflect the values in the **VERASE** and **VKILL** members of the **c_cc** field of the **termios** structure.

The **sg_flags** field of the argument structure contains several flags that determine the system's treatment of the terminal. They are mapped into flags in fields of the terminal state, represented by the **termios** structure.

Delay type **0** is always mapped into the equivalent delay type **0** in the **c_oflag** field of the **termios** structure. Other delay mappings are performed as follows:

sg_flags	**c_oflag**
BS1	**BS1**
FF1	**VT1**
CR1	**CR2**
CR2	**CR3**
CR3	not supported
TAB1	**TAB1**
TAB2	**TAB2**
XTABS	**TAB3**
NL1	**ONLRET|CR1**
NL2	**NL1**

If previous **TIOCLSET** or **TIOCLBIS ioctl** calls have not selected **LITOUT** or **PASS8** mode, and if **RAW** mode is not selected, then the **ISTRIP** flag is set in the **c_iflag** field of the **termios** structure, and the **EVENP** and **ODDP** flags control the parity of characters sent to the terminal and accepted from the terminal.

Parity is not to be generated on output or checked on input. The character size is set to **CS8** and the flag is cleared in the **c_cflag** field of the **termios** structure.

Even parity characters are to be generated on output and accepted on input. The flag is set in the **c_iflag** field of the **termios** structure, the character size is set to **CS7** and the flag is set in the **c_cflag** field of the **termios** structure.

Odd parity characters are to be generated on output and accepted on input. The flag is set in the **c_iflag** field, the character size is set to **CS7** and the flags are set in the **c_cflag** field of the **termios** structure.

Even parity characters are to be generated on output and characters of either parity are to be accepted on input. The flag is cleared in the **c_iflag** field, the character size is set to **CS7** and the flag is set in the **c_cflag** field of the **termios** structure.

The **RAW** flag disables all output processing (the **OPOST** flag in the **c_oflag** field, and the **XCASE** flag in the **c_lflag** field, are cleared in the **termios** structure) and input processing (all flags in the **c_iflag** field other than the **IXOFF** and **IXANY** flags are cleared in the **termios** structure). 8 bits of data, with no parity bit, are accepted on input and generated on output; the character size is set to **CS8** and the **PARENB** and **PARODD** flags are cleared in the **c_cflag** field of the **termios** structure. The signal-generating and line-editing control characters are disabled by clearing the **ISIG** and **ICANON** flags in the **c_lflag** field of the **termios** structure.

The **CRMOD** flag turns input RETURN characters into NEWLINE characters, and output and echoed NEWLINE characters to be output as a RETURN followed by a LINEFEED. The **ICRNL** flag in the **c_iflag** field, and the **OPOST** and **ONLCR** flags in the **c_oflag** field, are set in the **termios** structure.

The **LCASE** flag maps upper-case letters in the ASCII character set to their lower-case equivalents on input (the **IUCLC** flag is set in the **c_iflag** field), and maps lower-case letters in the ASCII character set to their upper-case equivalents on output (the **OLCUC** flag is set in the **c_oflag** field). Escape sequences are accepted on input, and generated on output, to handle certain ASCII characters not supported by older terminals (the **XCASE** flag is set in the **c_lflag** field).

Other flags are directly mapped to flags in the **termios** structure:

sg_flags	flags in **termios** structure
CBREAK	complement of **ICANON** in **c_lflag** field
ECHO	**ECHO** in **c_lflag** field
TANDEM	**IXOFF** in **c_iflag** field

Another structure associated with each terminal specifies characters that are special in both the old Version 7 and the newer **4BSD** terminal interfaces. The following structure is defined by **<sys/ttold.h>**:

```
struct   tchars  {
        char      t_intrc;     /* interrupt */
        char      t_quitc;     /* quit */
        char      t_startc;    /* start output */
        char      t_stopc;     /* stop output */
        char      t_eofc;      /* end-of-file */
        char      t_brkc;      /* input delimiter (like nl) */
};
```

XENIX defines the **tchar** structure as **tc**. The characters are mapped to members of the **c_cc** field of the **termios** structure as follows:

tchars	**c_cc** index
t_intrc	**VINTR**
t_quitc	**VQUIT**
t_startc	**VSTART**
t_stopc	**VSTOP**
t_eofc	**VEOF**
t_brkc	**VEOL**

Also associated with each terminal is a local flag word, specifying flags supported by the new **4BSD** terminal interface. Most of these flags are directly mapped to flags in the **termios** structure:

local flags	flags in **termios** structure
LCRTBS	not supported
LPRTERA	**ECHOPRT** in the **c_lflag** field
LCRTERA	**ECHOE** in the **c_lflag** field
LTILDE	not supported
LTOSTOP	**TOSTOP** in the **c_lflag** field
LFLUSHO	**FLUSHO** in the **c_lflag** field
LNOHANG	**CLOCAL** in the **c_cflag** field
LCRTKIL	**ECHOKE** in the **c_lflag** field
LCTLECH	**CTLECH** in the **c_lflag** field
LPENDIN	**PENDIN** in the **c_lflag** field
LDECCTQ	complement of **IXANY** in the **c_iflag** field
LNOFLSH	**NOFLSH** in the **c_lflag** field

Another structure associated with each terminal is the **ltchars** structure which defines control characters for the new **4BSD** terminal interface. Its structure is:

```
struct ltchars {
        char    t_suspc;     /* stop process signal */
        char    t_dsuspc;    /* delayed stop process signal */
        char    t_rprntc;    /* reprint line */
        char    t_flushc;    /* flush output (toggles) */
        char    t_werasc;    /* word erase */
        char    t_lnextc;    /* literal next character */
};
```

The characters are mapped to members of the **c_cc** field of the **termios** structure as follows:

ltchars	**c_cc** index
t_suspc	**VSUSP**
t_dsuspc	**VDSUSP**
t_rprntc	**VREPRINT**
t_flushc	**VDISCARD**
t_werasc	**VWERASE**
t_lnextc	**VLNEXT**

IOCTLS

ttcompat responds to the following **ioctl** calls. All others are passed to the module below.

TIOCGETP The argument is a pointer to an **sgttyb** structure. The current terminal state is fetched; the appropriate characters in the terminal state are stored in that structure, as are the input and output speeds. The values of the flags in the **sg_flags** field are derived from the flags in the terminal state and stored in the structure.

TIOCEXCL Set ''exclusive-use'' mode; no further opens are permitted until the file has been closed.

TIOCNXCL Turn off ''exclusive-use'' mode.

TIOCSETP The argument is a pointer to an **sgttyb** structure. The appropriate characters and input and output speeds in the terminal state are set from the values in that structure, and the flags in the terminal state are set to match the values of the flags in the **sg_flags** field of that structure. The state is changed with a **TCSETSF ioctl** so that the interface delays until output is quiescent, then throws away any unread characters, before changing the modes.

TIOCSETN The argument is a pointer to an **sgttyb** structure. The terminal state is changed as **TIOCSETP** would change it, but a **TCSETS ioctl** is used, so that the interface neither delays nor discards input.

TIOCHPCL The argument is ignored. The **HUPCL** flag is set in the **c_cflag** word of the terminal state.

TIOCFLUSH The argument is a pointer to an **int** variable. If its value is zero, all characters waiting in input or output queues are flushed. Otherwise, the value of the **int** is treated as the logical **OR** of the **FREAD** and **FWRITE** flags defined by **<sys/file.h>**. If the **FREAD** bit is set, all characters waiting in input queues are flushed, and if the **FWRITE** bit is set, all characters waiting in output queues are flushed.

TIOCBRK The argument is ignored. The break bit is set for the device.

TIOCCBRK The argument is ignored. The break bit is cleared for the device.

TIOCSDTR The argument is ignored. The Data Terminal Ready bit is set for the device.

TIOCCDTR The argument is ignored. The Data Terminal Ready bit is cleared for the device.

TIOCSTOP The argument is ignored. Output is stopped as if the STOP character had been typed.

TIOCSTART The argument is ignored. Output is restarted as if the START character had been typed.

TIOCGETC The argument is a pointer to a **tchars** structure. The current terminal state is fetched, and the appropriate characters in the terminal state are stored in that structure.

TIOCSETC The argument is a pointer to a **tchars** structure. The values of the appropriate characters in the terminal state are set from the characters in that structure.

TIOCLGET The argument is a pointer to an **int**. The current terminal state is fetched, and the values of the local flags are derived from the flags in the terminal state and stored in the **int** pointed to by the argument.

TIOCLBIS The argument is a pointer to an **int** whose value is a mask containing flags to be set in the local flags word. The current terminal state is fetched, and the values of the local flags are derived from the flags in the terminal state; the specified flags are set, and the flags in the terminal state are set to match the new value of the local flags word.

TIOCLBIC The argument is a pointer to an **int** whose value is a mask containing flags to be cleared in the local flags word. The current terminal state is fetched, and the values of the local flags are derived from the flags in the terminal state; the specified flags are cleared, and the flags in the terminal state are set to match the new value of the local flags word.

TIOCLSET The argument is a pointer to an **int** containing a new set of local flags. The flags in the terminal state are set to match the new value of the local flags word.

TIOCGLTC The argument is a pointer to an **ltchars** structure. The values of the appropriate characters in the terminal state are stored in that structure.

TIOCSLTC The argument is a pointer to an **ltchars** structure. The values of the appropriate characters in the terminal state are set from the characters in that structure.

FIORDCHK Returns the number of immediately readable characters. The argument is ignored.

FIONREAD Returns the number of immediately readable characters in the **int** pointed to by the argument.

LDSMAP Calls the function **emsetmap** (*tp*, *mp*) if the function is configured in the kernel.

LDGMAP Calls the function **emgetmap** (*tp*, *mp*) if the function is configured in the kernel.

LDNMAP Calls the function **emunmap** (*tp*, *mp*) if the function is configured in the kernel.

The following **ioctls** are returned as successful for the sake of compatibility. However, nothing significant is done (that is, the state of the terminal is not changed in any way).

TIOCSETD	**LDOPEN**
TIOCGETD	**LDCLOSE**
DIOCSETP	**LDCHG**
DIOCSETP	**LDSETT**
DIIOGETP	**LDGETT**

SEE ALSO

ioctl(2), **termios**(3), **ldterm**(7M), **termio**(7I)

NOTES

TIOCBRK and **TIOCCBRK** should be handled by the driver. **FIONREAD** and **FIORDCHK** are handled in the stream head.