IEEE Standards Interpretation for IEEE Std 1003.1™-1990 IEEE Standard for Information Technology--Portable Operating System Interfaces (POSIX®)

Interpretation Request #82
Topic: cfgetispeed and cfsetispeed **Relevant Clauses:** 7.1.3.2

Clause 7.1.3.2 defines the beavior of the cgetispeed and cfsetispeed functions, as follows: | The cfsetispeed() function shall set the input baud rate stored in | the termios structure to which termios_p points. | The cfgetispeed() function shall return the input baud rate stored in | the termios structure to which termios_p points. From this, one would expect that the following code would result in the value of speed being set to X, if none of the function calls return -1. | if (cfsetispeed(&termios_p, X) == -1) | printf ("cfsetispeed error %d\n", errno); | if ((speed = cfgetispeed (&termios_p)) == -1) | printf ("cfgetispeed error %d\n", errno); Now, 7.1.3.2 also says | ... It is unspecified whether these return an error if an unsupported | baud rate is set.

So, one might conjecture that cfsetispeed or cfgetispeed might quietly fail for some values of X, but this behavior is only allowed if the baud rate X is "unsupported". The term "unsupported baud rate" is also used in 7.2.1.4, which says that tcsetarr() shall return -1 and set errno to EINVAL if "...an attempt was made to change an attribute represented in the termios structure to an unsupported value." Thus, if the baud rate X is unsupported, the following call to tcsetattr should return -1, if the input baud rate in termios_p is X. | if (tcsetattr(2, TCSANOW, &termios_p) == -1) | printf ("tcsetattr error %d\n", errno); In particular, consider the following program. | #define _POSIX_SOURCE | #include | #include | #include | | void main(){ | speed_t speed = 99; | struct termios termios_p; | printf("B0: %d\n", B0); | if (tcgetattr(2, &termios_p) == -1) | printf ("tcgetattr error %d\n", errno); | if ((speed = cfgetispeed (&termios_p)) == -1) | printf ("cfgetispeed error %d\n", errno); | printf("speed: %d\n", speed); | if (tcsetattr(2, TCSANOW, &termios_p) == -1) | printf ("tcsetattr error %d\n", errno); | if (cfsetispeed(&termios_p, B0) == -1) | printf ("cfsetispeed error %d\n", errno); | if ((speed = cfgetispeed (&termios_p)) == -1) | printf ("cfgetispeed error %d\n", errno); | printf("speed: %d\n", speed);

| }

Working through it step by step, we have: | #define _POSIX_SOURCE | #include | #include | #include | | void main(){ | speed_t speed = 99; | struct termios termios_p; | printf("B0: %d\n", B0); If this puts out "B0 : 0", we know B0 = 0. | if (tcgetattr(2, &termios_p) == -1) | printf ("tcgetattr error %d\n", errno); If this succeeds, we know file-descriptor 2 is valid for use with tcgetattr. | if ((speed = cfgetispeed (&termios_p)) == -1) | printf ("cfgetispeed error %d\n", errno); If this returns normally we know speed contains the input baud rate stored in termios_p, or else that baud rate value is unsupported. | printf("speed: %d\n", speed); If this prints out "speed: 0", we know the input baud rate stored in termios_p is 0. | if (tcsetattr(2, TCSANOW, &termios_p) == -1) | printf ("tcsetattr error %d\n", errno); If this returns normally, we know that the values specified in termios_p are supported. | if (cfsetispeed(&termios_p, B0) == -1) | printf ("cfsetispeed error %d\n", errno); If this returns normally we know the input baud rate value stored in termios_p is B0, or else that baud rate value is unsupported. | if ((speed = cfgetispeed (&termios_p)) == -1) | printf ("cfgetispeed error %d\n", errno); If this returns normally we know speed contains the input baud rate value stored in termios_p, or else that baud rate value is unsupported. | printf("speed: %d\n", speed); This should print out "speed: 0" if all of the calls above succeeded, and B0 = 0, and 0 is a supported baud rate value. | } In other words, the following output would be incorrect. B0: 0 speed: 0 speed: 13

Note: This interpretation request also applies to 1003.1-1993 and 1003.1-1995, as the specifications in question do not appear to have changed between versions.

Interpretation Response
The implied question is, is an implementation that returns these values conforming. The response is yes. The standard clearly states that B0 is a special value, and that the value set in the structure by cfsetispeed() is not required to be returned "as is" by a call to cfgetispeed().

Rationale for Interpretation
None.