

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

Copyright © 1998 by the Institute of Electrical and Electronics Engineers, Inc. 3 Park Avenue New York, New York 10016-5997 USA All Rights Reserved.

Interpretations are issued to explain and clarify the intent of a standard and do not constitute an alteration to the original standard. In addition, interpretations are not intended to supply consulting information. Permission is hereby granted to download and print one copy of this document. Individuals seeking permission to reproduce and/or distribute this document in its entirety or portions of this document must contact the IEEE Standards Department for the appropriate license. Use of the information contained in this document is at your own risk.

IEEE Standards Department Copyrights and Permissions 445 Hoes Lane, Piscataway, New Jersey 08855-1331, USA

Interpretation Request #81

Topic: `tcsetattr` **Relevant Clauses:** 7.2.1.2

Does POSIX.1 require that `tcsetattr(fildes, TCSANOW, termios_p)` preserve data that may have been received by the serial communications port but not yet processed and placed in the input queue? My reading of the Standard is that POSIX.1 is silent as to whether the data are preserved. This means that the effect on unprocessed input is unspecified, so a conforming implementation can discard the data. The critical specifications are:

- (1) If `optional_actions` is `TCSANOW`, the change shall occur immediately.
- (2) If `optional_actions` is `TCSADRAIN`, the change shall occur after all output written to `fildes` has been transmitted. This function should be used when changing parameters that affect output.
- (3) If `optional_actions` is `TCSAFLUSH`, the change shall occur after all output written to the object referred to by `fildes` has been transmitted, and all input that has been received, but not read, shall be discarded before the change is made. (1002.1-1990 p. 143, 144; subclause 7.2.1.2)

(Questions in the elaboration below are meant to be rhetorical. They are not meant to request specific interpretations.) The descriptions focus largely on output. This is because output is under the system's control to a much greater extent than is input. The description of `tcflow()`, for example, has a corresponding asymmetry, in that the system is required to directly control output flow but only to send START and STOP out on the serial line to request that the remote system control input flow. Consider the case of a serial port that has an input buffer that holds raw input, to await processing in batches.

This is an attractive architecture because it can greatly reduce the number of interrupts

needed to service input on the port. What should happen to this buffered raw input when `tcsetattr(fildes, TCSANOW, termios_p)` is called? If the newly-set parameters include any changes to the input processing mode, the result of input processing on a system with such buffers could be different from what would be placed in the input queue by an implementation that does not buffer unprocessed data. Things are even worse for a character that has been only partly received from the transmission line by the serial hardware when the interface is reset. There is no unique correct way to process such data, and in many situations the data will be irretrievably corrupted. My feeling is that the TCSANOW optional action means “Make the change right now, no matter what happens to the data in transit”. Tools are provided to allow the programmer to trade responsiveness for data integrity for output data. No such facilities and no corresponding guarantees are provided for input, because the problem is fundamentally difficult. It’s impossible for the system to know what is about to arrive from the incoming serial connection.

Interpretation Response

The standard is silent on this issue, as to whether the data are preserved , and as such no conformance distinction can be made between different implementations.

Rationale for Interpretation

None.