

## IEEE Standards Interpretation for IEEE Std 1003.1™-2001 IEEE Standard Standard for Information Technology -- Portable Operating System Interface (POSIX®)

Copyright © 2006 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 #53

**Topic:** `asctime()` `tm_year` `gt 9999` **Relevant Sections:** XSH `asctime()` Page: 123 Line: 4407-4428

If `asctime()` is called with a `tm` structure whose `tm_year` field results in a year `> 9999` (which is possible with 64-bit `time_t`), the current specification of `asctime()` would result in `asctime()` to overrunning a 26-character buffer; the specification says the `sprintf()` format for printing the year is `"%d"`, and (eg) a 5-digit number would print 5 characters, overrunning the buffer.

Similarly, since the user can create the input struct `tm`, it is possible for the user to set the fields of the struct `tm` to values that are outside the normal bounds. In such a case, the `sprintf()` format given in the `asctime()` specification can result in a buffer overrun. For example, if `tm_hour` is 100, the `sprintf()` format `".2d"` writes the string `"100"`, which could result in a buffer overrun. The specification should be updated to state the algorithm can be used as long as the values of the `tm` struct are restricted to the normal bounds.

(Note TC2 already says `asctime()` can return `NULL` if unsuccessful.)

11 Solution proposed by the submitter (optional):

Action:

page 123 line 4487-4488 section `asctime()` objection

Change: The `asctime()` function shall convert the broken-down time in the structure pointed to by `timeptr` into a string in the form to The `asctime()` function shall convert the broken-down time in the structure pointed to by `timeptr` into a string in the form, pro-

vided the broken-down time in the fields of the structure pointed to by timeptr contain values that are within the normal ranges (see ), and the calculated year does not exceed four digits:

page 123 line 4508 section asctime() objection

After line 4508, add:

Otherwise, if any of the fields of the tm structure pointed to by timeptr contain values that are outside the normal ranges, asctime() behavior is undefined. If the calculated year exceeds four digits, asctime() shall either return NULL and set errno to EOVERFLOW, or write no more than four unspecified characters to the year field.

page 123 line 4526 section asctime() objection

Change: No errors are defined to The asctime() and asctime\_r() functions may fail if: [EOVERFLOW] The result cannot be represented.

### **Interpretation Response #53**

The standards states the requirements for asctime(), and conforming implementations must conform to this. However, concerns have been raised about this which are being referred to the sponsor.”

### **Rationale for Interpretation**

None.