

## 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 #57

**Topic:** pthread\_attr\_getstackaddr, stackaddr attribute unset **Relevant Sections:** XSH pthread\_attr\_getstackaddr Page: 1012 Line: 32362

There appears to be differing behavior amongst implementations for what happens on the following call sequence:

```
rval = pthread_attr_init( &attr );  
...  
rval = pthread_attr_getstackaddr( &attr, &addr );
```

The description of pthread\_attr\_init states: "The pthread\_attr\_init() function shall initialize a thread attributes object attr with the default value for all of the individual attributes used by a given implementation."

On many implementations the call to pthread\_attr\_getstackaddr() succeeds.

One implementation we have recently come up against returns rval == EINVAL for this case and will only succeed if there has been a prior call to pthread\_attr\_setstackaddr(). (1)Is this alternative behavior conforming?

(2)If conforming, is the use of EINVAL appropriate given that TC2 has added an EINVAL for invalid attributes and the attributes in this case are valid due to the prior call to pthread\_attr\_init().

If the behavior is deemed conforming, consider revising the wording to note that a call to pthread\_attr\_getstackaddr() may fail without a prior call to pthread\_attr\_setstackaddr()

and add an appropriate error condition.

**Interpretation Response #57**

The standard does not speak to this issue, and as such no conformance distinction can be made between alternative implementations based on this. This is being referred to the sponsor.

**Rationale for Interpretation**

None.