



Red Hat Enterprise Linux Release 9.2 Manual Pages on 'pthread_getspecific.3p' command

\$ man pthread_getspecific.3p

PTHREAD_GETSPECIFIC(3P) POSIX Programmer's Manual PTHREAD_GETSPECIFIC(3P)

PROLOG

This manual page is part of the POSIX Programmer's Manual. The Linux implementation of this interface may differ (consult the corresponding Linux manual page for details of Linux behavior), or the interface may not be implemented on Linux.

NAME

pthread_getspecific, pthread_setspecific ? thread-specific data management

SYNOPSIS

```
#include <pthread.h>

void *pthread_getspecific(pthread_key_t key);

int pthread_setspecific(pthread_key_t key, const void *value);
```

DESCRIPTION

The `pthread_getspecific()` function shall return the value currently bound to the specified key on behalf of the calling thread.

The `pthread_setspecific()` function shall associate a thread-specific value with a key obtained via a previous call to `pthread_key_create()`.

Different threads may bind different values to the same key. These values are typically pointers to blocks of dynamically allocated memory that have been reserved for use by the calling thread.

The effect of calling `pthread_getspecific()` or `pthread_setspecific()` with a key value not obtained from `pthread_key_create()` or after key

has been deleted with `pthread_key_delete()` is undefined.

Both `pthread_getspecific()` and `pthread_setspecific()` may be called from a thread-specific data destructor function. A call to `pthread_getspecific()` for the thread-specific data key being destroyed shall return the value `NULL`, unless the value is changed (after the destructor starts) by a call to `pthread_setspecific()`. Calling `pthread_setspecific()` from a thread-specific data destructor routine may result either in lost storage (after at least `PTHREAD_DESTRUCTOR_ITERATIONS` attempts at destruction) or in an infinite loop.

Both functions may be implemented as macros.

RETURN VALUE

The `pthread_getspecific()` function shall return the thread-specific data value associated with the given key. If no thread-specific data value is associated with key, then the value `NULL` shall be returned.

If successful, the `pthread_setspecific()` function shall return zero; otherwise, an error number shall be returned to indicate the error.

ERRORS

No errors are returned from `pthread_getspecific()`.

The `pthread_setspecific()` function shall fail if:

`ENOMEM` Insufficient memory exists to associate the non-`NULL` value with the key.

The `pthread_setspecific()` function shall not return an error code of `[EINTR]`.

The following sections are informative.

EXAMPLES

None.

APPLICATION USAGE

None.

RATIONALE

Performance and ease-of-use of `pthread_getspecific()` are critical for functions that rely on maintaining state in thread-specific data. Since no errors are required to be detected by it, and since the only error that could be detected is the use of an invalid key, the function to

pthread_getspecific() has been designed to favor speed and simplicity over error reporting.

If an implementation detects that the value specified by the key argument to pthread_setspecific() does not refer to a key value obtained from pthread_key_create() or refers to a key that has been deleted with pthread_key_delete(), it is recommended that the function should fail and report an [EINVAL] error.

FUTURE DIRECTIONS

None.

SEE ALSO

pthread_key_create()

The Base Definitions volume of POSIX.1-2017, <pthread.h>

COPYRIGHT

Portions of this text are reprinted and reproduced in electronic form from IEEE Std 1003.1-2017, Standard for Information Technology -- Portable Operating System Interface (POSIX), The Open Group Base Specifications Issue 7, 2018 Edition, Copyright (C) 2018 by the Institute of Electrical and Electronics Engineers, Inc and The Open Group. In the event of any discrepancy between this version and the original IEEE and The Open Group Standard, the original IEEE and The Open Group Standard is the referee document. The original Standard can be obtained online at <http://www.opengroup.org/unix/online.html>.

Any typographical or formatting errors that appear in this page are most likely to have been introduced during the conversion of the source files to man page format. To report such errors, see https://www.kernel.org/doc/man-pages/reporting_bugs.html.

IEEE/The Open Group 2017 PTHREAD_GETSPECIFIC(3P)