Full credit is given to the above companies including the OS that this PDF file was generated!

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

$ man pthread_exit.3p

PTHREAD_EXIT(3P) POSIX Programmer's Manual PTHREAD_EXIT(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_exit ? thread termination

SYNOPSIS

#include <pthread.h>

void pthread_exit(void *value_ptr);

DESCRIPTION

The pthread_exit() function shall terminate the calling thread and make the value value_ptr available to any successful join with the terminat? ing thread. Any cancellation cleanup handlers that have been pushed and not yet popped shall be popped in the reverse order that they were pushed and then executed. After all cancellation cleanup handlers have been executed, if the thread has any thread-specific data, appropriate destructor functions shall be called in an unspecified order. Thread termination does not release any application visible process resources, including, but not limited to, mutexes and file descriptors, nor does it perform any process-level cleanup actions, including, but not lim? ited to, calling any atexit() routines that may exist.

An implicit call to pthread_exit() is made when a thread other than the thread in which main() was first invoked returns from the start routine that was used to create it. The function's return value shall serve as the thread's exit status.

The behavior of pthread_exit() is undefined if called from a cancella? tion cleanup handler or destructor function that was invoked as a re? sult of either an implicit or explicit call to pthread_exit().

After a thread has terminated, the result of access to local (auto) variables of the thread is undefined. Thus, references to local vari? ables of the exiting thread should not be used for the pthread_exit() value_ptr parameter value.

The process shall exit with an exit status of 0 after the last thread has been terminated. The behavior shall be as if the implementation called exit() with a zero argument at thread termination time.

RETURN VALUE

The pthread_exit() function cannot return to its caller.

ERRORS

No errors are defined.

The following sections are informative.

EXAMPLES

None.

APPLICATION USAGE

None.

RATIONALE

The normal mechanism by which a thread terminates is to return from the routine that was specified in the pthread_create() call that started it. The pthread_exit() function provides the capability for a thread to terminate without requiring a return from the start routine of that thread, thereby providing a function analogous to exit().

Regardless of the method of thread termination, any cancellation cleanup handlers that have been pushed and not yet popped are executed, and the destructors for any existing thread-specific data are executed.

This volume of POSIX.1?2017 requires that cancellation cleanup handlers

be popped and called in order. After all cancellation cleanup handlers have been executed, thread-specific data destructors are called, in an unspecified order, for each item of thread-specific data that exists in the thread. This ordering is necessary because cancellation cleanup handlers may rely on thread-specific data.

As the meaning of the status is determined by the application (except when the thread has been canceled, in which case it is PTHREAD_CAN? CELED), the implementation has no idea what an illegal status value is, which is why no address error checking is done.

FUTURE DIRECTIONS

None.

SEE ALSO

exit(), pthread_create(), pthread_join()

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