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_cleanup_push_defer_np.3' command

$ man pthread_cleanup_push_defer_np.3

PTHREAD_CLEANUP_PUSH_DEFER_Linux Programmer's PTHREAD_CLEANUP_PUSH_DEFER_NP(3)

NAME

pthread_cleanup_push_defer_np, pthread_cleanup_pop_restore_np - push and pop thread cancellation clean-up handlers while saving cancelabil? ity type

SYNOPSIS

```
#include <pthread.h>
void pthread_cleanup_push_defer_np(void (*routine)(void *),
                                   void *arg);
void pthread_cleanup_pop_restore_np(int execute);
```

Compile and link with -pthread.

Feature Test Macro Requirements for glibc (see feature_test_macros(7)):

pthread_cleanup_push_defer_np(), pthread_cleanup_pop_defer_np():

_GNU_SOURCE

DESCRIPTION

These functions are the same as pthread_cleanup_push(3) and pthread_cleanup_pop(3), except for the differences noted on this page.

Like pthread_cleanup_push(3), pthread_cleanup_push_defer_np() pushes routine onto the thread's stack of cancellation clean-up handlers. In addition, it also saves the thread's current cancelability type, and sets the cancelability type to "deferred" (see pthread_setcancel? type(3)); this ensures that cancellation clean-up will occur even if the thread's cancelability type was "asynchronous" before the call.

Like pthread_cleanup_pop(3), pthread_cleanup_pop_restore_np() pops the top-most clean-up handler from the thread's stack of cancellation clean-up handlers. In addition, it restores the thread's cancelability type to its value at the time of the matching pthread_cleanup_push_de? fer_np().

The caller must ensure that calls to these functions are paired within the same function, and at the same lexical nesting level. Other re? strictions apply, as described in pthread_cleanup_push(3).

This sequence of calls:

```
pthread_cleanup_push_defer_np(routine, arg);
pthread_cleanup_pop_restore_np(execute);
```

is equivalent to (but shorter and more efficient than):

```
int oldtype;
pthread_cleanup_push(routine, arg);
pthread_setcanceltype(PTHREAD_CANCEL_DEFERRED, &oldtype);
...
pthread_setcanceltype(oldtype, NULL);
pthread_cleanup_pop(execute);
```

CONFORMING TO

These functions are nonstandard GNU extensions; hence the suffix "_np" (nonportable) in the names.

SEE ALSO

pthread_cancel(3), pthread_cleanup_push(3), pthread_setcancelstate(3), pthread_testcancel(3), pthreads(7)

COLOPHON

This page is part of release 5.10 of the Linux man-pages project. A description of the project, information about reporting bugs, and the latest version of this page, can be found at https://www.kernel.org/doc/man-pages/.