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_setaffinity_np.3' command

$ man pthread_setaffinity_np.3

PTHREAD_SETAFFINITY_NP(3) Linux Programmer's Manual PTHREAD_SETAFFINITY_NP(3)

NAME

pthread_setaffinity_np, pthread_getaffinity_np - set/get CPU affinity of a thread

SYNOPSIS

```
#define _GNU_SOURCE             /* See feature_test_macros(7) */
#include <pthread.h>
int pthread_setaffinity_np(pthread_t thread, size_t cpusetsize,
                           const cpu_set_t *cpuset);
int pthread_getaffinity_np(pthread_t thread, size_t cpusetsize,
                           cpu_set_t *cpuset);
```

Compile and link with -pthread.

DESCRIPTION

The pthread_setaffinity_np() function sets the CPU affinity mask of the thread thread to the CPU set pointed to by cpuset. If the call is suc? cessful, and the thread is not currently running on one of the CPUs in cpuset, then it is migrated to one of those CPUs.

The pthread_getaffinity_np() function returns the CPU affinity mask of the thread thread in the buffer pointed to by cpuset.

For more details on CPU affinity masks, see sched_setaffinity(2). For a description of a set of macros that can be used to manipulate and in? spect CPU sets, see CPU_SET(3).

The argument cpusetsize is the length (in bytes) of the buffer pointed

to by cpuset. Typically, this argument would be specified as sizeof(cpu_set_t). (It may be some other value, if using the macros described in CPU_SET(3) for dynamically allocating a CPU set.)

RETURN VALUE

On success, these functions return 0; on error, they return a nonzero error number.

ERRORS

EFAULT A supplied memory address was invalid.

EINVAL (pthread_setaffinity_np()) The affinity bit mask mask contains no processors that are currently physically on the system and permitted to the thread according to any restrictions that may be imposed by the "cpuset" mechanism described in cpuset(7).

EINVAL (pthread_setaffinity_np()) cpuset specified a CPU that was out? side the set supported by the kernel. (The kernel configuration option CONFIG_NR_CPUS defines the range of the set supported by the kernel data type used to represent CPU sets.)

EINVAL (pthread_getaffinity_np()) cpusetsize is smaller than the size of the affinity mask used by the kernel.

ESRCH No thread with the ID thread could be found.

VERSIONS

These functions are provided by glibc since version 2.3.4.

ATTRIBUTES

For an explanation of the terms used in this section, see at? tributes(7).

??

?Interface ? Attribute ? Value ?

??

?pthread_setaffinity_np(), ? Thread safety ? MT-Safe ?

?pthread_getaffinity_np() ? ? ?

??

CONFORMING TO

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

NOTES

After a call to pthread_setaffinity_np(), the set of CPUs on which the thread will actually run is the intersection of the set specified in the cpuset argument and the set of CPUs actually present on the system. The system may further restrict the set of CPUs on which the thread runs if the "cpuset" mechanism described in cpuset(7) is being used. These restrictions on the actual set of CPUs on which the thread will run are silently imposed by the kernel.

These functions are implemented on top of the sched_setaffinity(2) and sched_getaffinity(2) system calls.

In glibc 2.3.3 only, versions of these functions were provided that did not have a cpusetsize argument. Instead the CPU set size given to the underlying system calls was always sizeof(cpu_set_t).

A new thread created by pthread_create(3) inherits a copy of its cre? ator's CPU affinity mask.

EXAMPLES

In the following program, the main thread uses pthread_setaffinity_np() to set its CPU affinity mask to include CPUs 0 to 7 (which may not all be available on the system), and then calls pthread_getaffinity_np() to check the resulting CPU affinity mask of the thread.

```
#define _GNU_SOURCE
#include <pthread.h>
#include <stdio.h>
#include <stdlib.h>
#include <errno.h>
#define handle_error_en(en, msg) \
        do { errno = en; perror(msg); exit(EXIT_FAILURE); } while (0)
int
main(int argc, char *argv[])
{
    int s;
    cpu_set_t cpuset;
    pthread_t thread;
```

```
        thread = pthread_self();
        /* Set affinity mask to include CPUs 0 to 7 */
        CPU_ZERO(&cpuset);
        for (int j = 0; j < 8; j++)
            CPU_SET(j, &cpuset);
        s = pthread_setaffinity_np(thread, sizeof(cpuset), &cpuset);
        if (s != 0)
            handle_error_en(s, "pthread_setaffinity_np");
        /* Check the actual affinity mask assigned to the thread */
        s = pthread_getaffinity_np(thread, sizeof(cpuset), &cpuset);
        if (s != 0)
            handle_error_en(s, "pthread_getaffinity_np");
        printf("Set returned by pthread_getaffinity_np() contained:\n");
        for (int j = 0; j < CPU_SETSIZE; j++)
            if (CPU_ISSET(j, &cpuset))
                printf("    CPU %d\n", j);
        exit(EXIT_SUCCESS);
    }
```

SEE ALSO

sched_setaffinity(2), CPU_SET(3), pthread_attr_setaffinity_np(3), pthread_self(3), sched_getcpu(3), cpuset(7), pthreads(7), sched(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/.