**NAME**

thr_min_stack – returns the minimum-allowable size for a thread's stack

**SYNOPSIS**

**cc** [ *flag* ... ] *file* ... **–lthread** [ *library* ... ]

**#include <thread.h>**

**size_t thr_min_stack(***void***);**

**MT-LEVEL**

MT-Safe

**DESCRIPTION**

When a thread is created with a user-supplied stack, the user must reserve enough space to run this thread. In a dynamically linked execution environment, it is very hard to know what the minimum stack requirments are for a thread. The function **thr_min_stack()** returns the amount of space needed to execute a null thread. This is a thread that was created to execute a null procedure. A thread that does something useful should have a stack size that is **thr_min_stack()** + <*some increment*>.

Most users should not be creating threads with user-supplied stacks. This functionality was provided to support applications that wanted complete control over their execution environment.

Typically, users should let the threads library manage stack allocation. The threads library provides default stacks which should meet the requirements of any created thread.

**thr_min_stack( )** will return the unsigned int **THR_MIN_STACK,** which is the minimum-allowable size for a thread's stack.

In this implementation the default size for a user-thread's stack is one mega-byte. If the second argument to **thr_create**(3T) is NULL, then the default stack size for the newly-created thread will be used. Otherwise, you may specify a stack-size that is at least **THR_MIN_STACK,** yet less than the size of your machine's virtual memory.

It is recommended that the default stack size be used.

To determine the smallest-allowable size for a thread's stack, execute the following:

```
/* cc thisfile.c -lthread */
#define _REENTRANT
#include <thread.h>
#include <stdio.h>

main() {
     printf("thr_min_stack() returns %u\n",thr_min_stack());
}
```

**SEE ALSO**

**pthread_attr_init**(3T), **pthread_create**(3T)

**NOTES**

Although the POSIX threads implementation, pthreads, does not have a corresponding function to **thr_min_stack( ),** it does implement a minimum stack size, whose value is **PTHREAD_STACK_MIN,** which may be ascertained as follows:

```
/* cc thisfile.c -lpthread */
#define _REENTRANT
#include <pthread.h>
#include <stdio.h>

main()  {
      printf("minimum POSIX stack size is  %u\n", PTHREAD_STACK_MIN);
}
```