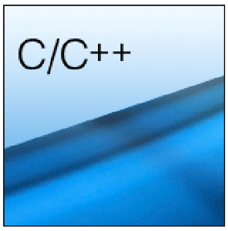
C/C++ Cafe

登录
以进行参与

C/C++ Cafe - Community & Forum



概述

最近更新

成员

博客

书签

Wiki

Don't forget about stack size limits for multi-threaded applications on AIX

BasilTK | *2013年9月7日* |
Comment (1) | *Visits (15124)*

1

G+1 190

Share

Tweet

Have you ever wondered why your child threads in a multi-threaded application on AIX fail to acquire a large amount of memory from the stack (even though your ulimits are set to unlimited) ?

The reason is simple. For a 32-bit compiled application on AIX, the default pthread stacksize is 96 KB; and for a 64-bit compiled application on AIX,
the default stacksize is 192 KB. Both of these default values are defined in the pthread.h header file.

Let's use the following test case as an example.
$ cat test.c

```
#include <stdio.h>

#include <pthread.h>


void *TestThread();


main(int Argc,char *Argv[])

{

   int    i;

   pthread_t tid;

11      //Main thread memory allocation

12      //char Buf[4096*30+1];

13      //memset(Buf, 0x00, sizeof(Buf));
14

15      pthread_create(&tid, NULL, TestThread, NULL);

16



17    }

18

19    void *TestThread()

20    {

21       int    i;

22       //Child thread memory allocation
23       char Buf[4096*30+1];

24       printf("sizeof Buf = %d \n",sizeof(Buf));
25       memset(Buf, 0x00, sizeof(Buf)); //<---Segfaults here

26       usleep(50);

27

28       while(1) sleep(1);

29

30     }

$
```

Line 25 of test.c tries to allocate (within the child thread) a buffer of size 122881 bytes on the stack; but it will segfault at runtime (as shown in the output below) because it is greater than 32 KB.

```
$ xlc_r test.c
$ ./a.out
sizeof Buf = 122881
Segmentation fault(coredump)
$
```

Now, if you comment out line 25 and un-comment lines 12 & 13, the test case will succeed at allocating it because the main parent thread uses the stack value that is set via the "ulimit -s" setting. ie, the ulimit stack size is only for the main thread.

You may now be wondering, "How can I allocate more than 32 KB for the child threads?".

There are a few ways to achieve this:
1. Compile your application with -q64 instead of -q32. This will give you 192 KB of stack.

eg:

```
$ xlc_r test.c -q64
$ ./a.out
sizeof Buf = 122881
```

```
$
```

2. Use "malloc()" to allocate the required memory on the heap instead of the stack.

eg:

```
//Replace Line 23 & 25 in test.c
with the following respectively:
char* Buf2;
Buf2 = (char*)
malloc((4096*30+1) *
sizeof(char));
```

3. Use the "AIXTHREAD_STK" environment variable to set a bigger pthread stack size

eg:

```
$ export AIXTHREAD_STK=13
0000;
$ ./a.out
sizeof Buf = 122881
$
```

Note - More info on AIXTHREAD_STK can be obtained here: http://pic.dhe.ibm.com/infocenter/aix/v7r1/index.jsp?topic=%2Fcom.ibm.aix.prftungd%2Fdoc%2Fprftungd%2Fthread_supp_tun_params.htm

4. Use "pthread_attr_setstacksize" to set bigger pthread stack.

eg:

```
//The modified test case would
look something like:
$ cat test_mod.c
#include <stdio.h>


#include <pthread.h>

void *TestThread();

main(int Argc,char *Argv[])
{
        int     i;
        pthread_t tid;
        pthread_attr_t attr;
        size_t  stacksize = 0;

        pthread_attr_init( &attr );
        pthread_attr_getstacksize(
&attr, &stacksize );
        printf("before stacksize :
[%d]\n", stacksize);
        pthread_attr_setstacksize(
&attr, 99800000 );
        pthread_attr_getstacksize(
&attr, &stacksize );
        printf("after  stacksize :
[%d]\n", stacksize);

        pthread_create(&tid, &attr,
TestThread, NULL);

}
void *TestThread()
{
        int     i;
        char Buf[4096*30+1];
```