gcc的 `__thread` keyword用于定义tls (Thread Local Storage) variable。

```c
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>

#define THREADS 3

__thread int tls = -1;
int global;

void *func(void *arg)
{
    int num = (int) arg;
    printf("Thread = %d tls = %d\n", num, tls);
    tls = num;
    global = num;
    sleep(1);
    printf("Thread = %d tls = %d global = %d\n", num, tls, global);
}

int main()
{
    int ret;
    pthread_t thread[THREADS];
    int num;

    for(num = 0; num < THREADS; num++)
    {
        ret = pthread_create(thread + num, NULL, func, (void *)num);
        if(ret)
        {
            printf("error pthread_create\n");
            exit(1);
        }
    }

    for(num = 0; num < THREADS; num++)
    {
        ret = pthread_join(thread[num], NULL);
        if(ret)
        {
            printf("error pthread_join\n");
            exit(2);
        }
    }
    exit(0);
}
```

> gcc -g -o thread_tls thread_tls.c -l pthread

```
walterzh@walterzh-Precision-T1650:~/work2/temp/tls$ ./thread_tls
Thread = 0 tls = -1
Thread = 1 tls = -1
Thread = 2 tls = -1
Thread = 0 tls = 0 global = 2
Thread = 1 tls = 1 global = 2
Thread = 2 tls = 2 global = 2
```

从 `tls` and `global` variables可以看出tls是每个thread都有的变量，而非全局变量。

```
    tls = num;
  4007dc:   8b 45 fc                mov    -0x4(%rbp),%eax
  4007df:   64 89 04 25 fc ff ff    mov    %eax,%fs:0xfffffffffffffffc
  4007e6:   ff
    global = num;
  4007e7:   8b 45 fc                mov    -0x4(%rbp),%eax
  4007ea:   89 05 7c 08 20 00       mov    %eax,0x20087c(%rip)        # 6010
6c <global>
```

在AMD-64上可以看出TLS variable放在fs所指向的空间

在ARM (32-bit)上

> arm-linux-gnueabi-gcc -g -o thread_tls thread_tls.c -l pthread

```
root@granite2:~# ./thread_tls
Thread = 1 tls = -1
Thread = 0 tls = -1
Thread = 2 tls = -1
Thread = 1 tls = 1 global = 2
Thread = 0 tls = 0 global = 2
Thread = 2 tls = 2 global = 2
```

```
    tls = num;
    864c:   eb00006f    bl  8810 <__aeabi_read_tp>
    8650:   e1a02000    mov r2, r0
    8654:   e59f3048    ldr r3, [pc, #72]   ; 86a4 <func+0x90>
    8658:   e51b1008    ldr r1, [fp, #-8]
    865c:   e7821003    str r1, [r2, r3]
```

> ldr r1, [fp, #-8]

这里[fp, #-8]是num

> str r1, [r2, r3]
> 把num的值(in r1 register)存入tls([r2, r3])

r2是__aeabi_read_tp function的返回值，应该是存放TLS variables的空间的base
而r3则是tls variable在TLS空间的offset。

```
00008810 <__aeabi_read_tp>:
    8810:   e3e00a0f    mvn r0, #61440  ; 0xf000
    8814:   e240f01f    sub pc, r0, #31
```

不太明白意思?