

errno多线程

多线程安全的errno

- C已经在与语言层面实现了thread local storage (TLS)，其中GNU C使用的就是__thread关键字

```
1  extern int *__errno_location (void) __THROW __attribute__ ((__const__));
2
3  extern __thread int errno attribute_tls_model_ie;
4 //glibc/csu/errno-loc.c
5 #include <errno.h>
6 #include <tls.h>
7
8 int *
9 __errno_location (void)
10 {
11     return &errno;
12 }
13 libc_hidden_def (__errno_location)
```

__thread关键字

- __thread是GCC内置的线程局部存储设施，其存储效率可以和全局变量相比；
- __thread变量在每一个线程中都有一份独立实例，各线程值是互不干扰的。可以用来修饰那些带有全局性且值可能变，但是又不值得用全局变量保护的变量。

限制

- 只能修饰POD类型（类似整型指针的标量，不带自定义的构造、拷贝、赋值、析构的类型，二进制内容可以任意复制memset,memcpy,且内容可以复原）。
- 不能修饰class类型，因为无法自动调用构造和析构函数。
- 可用于修饰全局变量，函数内的静态变量，不能修饰函数的局部变量或class的普通成员变量。且__thread变量值只能初始化为编译期常量
 - 这是因为线程局部存储需要在编译时分配内存，并且每个线程都需要拥有自己独立的初始值。
- __thread限定符(specifier)可以单独使用，也可带有extern或static限定符，但不能带有其它存储类型的限定符。
- __thread可用于全局的静态文件作用域，静态函数作用域或一个类中的静态数据成员。不能用于块作用域，自动或非静态数据成员。