

女朋友也能看懂的TLS技术讲解

原创 陈同学在搬砖 陈同学在搬砖

2020-02-21
00:04

FBI Warning

硬核预警：

本文需要一定的
多线程基础知识
没学过的小伙伴也没关系
本文将会通俗的讲解
说不定看着看着你就懂了呢

Q:

啥叫TLS？ 特仑苏？

A:

。 。 。 。

要知道啥叫**TLS**

先得知道它要用来解决啥问题

我们知道在一个进程中，
所有线程是共享同一个地址空间的。

所以，如果一个变量是全局变量，
那么所有线程访问的是同一份，

如果某一个线程对其进行了修改，
就会影响到其他所有的线程。
不过我们可能并不希望这样，

例如linux下的全局变量 `errno`，
线程A在设置了一个错误信息后，
线程B又设置了一个错误信息，
前一个线程设置的信息就被覆盖了。

那有没有办法保证在多线程程序中能访问而不互相影响呢？
答案是有的。

操作系统帮我们提供了这个功能

TLS(Thread Local Storage)线程本地存储。

TLS的作用是能将数据和执行的特定的线程联系起来。

所以上面的那个问题解决方法就是
将这个全局变量设置为TLS变量，

这样在用户看来虽然errno是一个全局变量，
但实际上它是每个线程独立的。
我们就是需要依赖全局变量或者静态变量，

Q:

还是没大明白..... 有没有课代表给总结一下

A:

总结一下就是

就是一个全局或者静态变量
但是把这个全局变量声明为TLS变量以后

它会在每个线程都有一个副本，
在不同的线程中可以按照相同的方式来访问，

既然是都有副本，自然线程中互不影响。

Q:

哦就好比一个人，被分身出了三个，其中一个被砍了一刀，其它两人都不会受伤。

A:

可以这样理解

Q:

TLS变量 = 影 流 之 主 ？

A:

WTF ？ ？ ？ ？

Q:

有没有严格的定义？

A:

TLS是一种在多线程时使用的技术，
它可以使你的**全局变量、静态变量以及局部静态、静态成员变量**
成为**线程独立的变量**，
即每个线程的TLS变量之间互不影响。
就像对待自身的局部变量一样。

Q:

emm.....光看理论还真是让人枯燥鸭.... Talk is cheap Show me the code

A:

得嘞..... 下面就教你怎么用

其实想用TLS 变量其实很简单

直接在全局变量声明前加个 `__thread`关键字就可以

像下面这样

Q:

只有这一种使用方法吗？

A:

其实上面只是最简单的一种使用方法

TLS的使用其实分为两种方式 静态的和动态的

静态的TLS

直接用语言本身的关键字去定义

在C/C++里面主要有这几个关键字

1. __thread
2. thread_local
3. __declspec (thread)

第三个关键字主要在visual c++编译器下使用
在g++/gcc编译器下用不了(实测)

使用方法和刚才说的一样

动态的TLS

调用系统的API去创建的 而不是用语言本身的关键字
而linux和windows调用的api又有所不同

可以自行查一下维基百科
这里不作赘述

Q:

老铁 有没有啥具体的例子 ???

A:

好滴 老板

给你两个

一看就懂的例子

没有加TLS

有加TLS

Q: 哦听你这么一说我大概会用了 ,那今天就到这吧

A:

少侠慢走!!!

学东西得知其然知其所以然

TLS这种强大的技术

难道你就不好奇它是如何实现的吗?

Q:

emm.....行吧 那你再给我唠唠它的原理

A:

线程局部存储在不同的平台有不同的实现，可移植性不太好。

幸好要实现线程局部存储并不难，

最简单的办法就是建立一个全局表，

通过当前线程ID去查询相应的数据，因为各个线程的ID不同，

查到的数据自然也就不同了。

但Windows系统采用了每个线程建线程专享的索引表，

表的条目为线程局部存储的地址。

在线程执行的任何代码处，

都可以查询本线程的这个索引表获得

要访问的线程局部存储的地址。

大多数平台都提供了线程局部存储的方法，

无需要我们自己去实现：

Q:

恩恩 通透现在算是知道他是用在哪 怎么用 啥原理了

A:

只要你明白了就好

下次有问题 可以后台私信我

我也要搬砖去了 bye~

对了

觉得有用

扫码关注

点个在看

你的支持是我搬砖的动力

我是陈同学

让技术

有温度