

线程局部变量的使用与多线程开发计算机等级考试 PDF转换  
可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/645/2021\\_2022\\_\\_E7\\_BA\\_BF\\_E7\\_A8\\_8B\\_E5\\_B1\\_80\\_E9\\_c97\\_645298.htm](https://www.100test.com/kao_ti2020/645/2021_2022__E7_BA_BF_E7_A8_8B_E5_B1_80_E9_c97_645298.htm)

多线程，现在多核时代来临之际，多线程编程已经成为了时髦语。有专门应运而生的Erlang，以及以Actor著称的Scala语言。下面介绍线程局部变量的使用，对于学习使用多线程会有一些帮助。一、概述  
现在多核时代多线程开发越来越重要了，多线程相比于多进程有诸多优势(当然也有诸多劣势)。在早期C的库中，有许多函数是线程不安全的，因为内部用到了静态变量，比如

：char \*strtok(char \*s, const char \*delim). 该函数内部就有一个静态指针，如果多个线程同时调用此函数时，可能就会出现奇怪的结果，当然也不是我们所想要的，现在Linux对此函数功能有一个线程安全版本的接口：char \*strtok\_r(char \*s, const char \*delim, char \*\*ptrptr)，这就避免了多个线程同时访问的冲突问题。其实，如果保持 strtok()/2 接口不变，同时还要保证线程安全，还有一个解决办法，那就是采用线程局部变量。使用线程局部变量有两种使用方式，一个稍微麻烦些，一个比较简单，下面一一做个介绍(以Linux为例) 二、线程局部变量的使用

比较麻烦些的使用方法用到的函数主要有三个：

pthread\_once(pthread\_once\_t\*, void

(\*init\_routine)(void)), pthread\_key\_create()/2,

pthread\_setspecific()/2, pthread\_getspecific()/1，其中

pthread\_once 可以保证在整个进程空间init\_routine函数仅被调用一次(它解决了多线程环境中使得互斥量和初始化代码都仅被初始化一次的问题).pthread\_key\_create 的参数之一指一个析

构造函数指针，当某个线程终止时该析构函数将被调用，并用  
对于一个进程内的给定键，该函数只能被调用一  
次。pthread\_sspecific 和 pthread\_getspecific 用来存放和获取与一个  
键关联的值。例子如下：  
pthread\_key\_t key; pthread\_once\_t  
once = PTHREAD\_ONCE\_INIT; static void destructor(void \*ptr)  
{ free(ptr); } void init\_once(void) { pthread\_key\_create( 100Test 下  
载频道开通，各类考试题目直接下载。详细请访问  
[www.100test.com](http://www.100test.com)