

Linux和Windows系统线程间的区别Linux认证考试 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/616/2021_2022_Linux_E5_92_8CWi_c103_616636.htm 熟悉WIN32编程的人一定知道

，WIN32的进程管理方式与Linux上有着很大区别，在Unix里，只有进程的概念，但在WIN32里却还有一个"线程"的概念，那么Linux和WIN32在这里究竟有着什么区别呢？WIN32里的进程/线程是继承自OS/2的。在WIN32里，"进程"是指一个程序，而"线程"是一个"进程"里的一个执行"线索"。从核心上讲，WIN32的多进程与Linux并无多大的区别，在WIN32里的线程才相当于Linux的进程，是一个实际正在执行的代码。但是，WIN32里同一个进程里各个线程之间是共享数据段的。

这才是与Linux的进程最大的不同。下面这段程序显示

了WIN32下一个进程如何启动一个线程。

```
int g; DWORD  
WINAPI ChildProcess( LPVOID lpParameter ){ int i; for ( i = 1; i  
amp;threadID ). for ( i = 1; i < 1000; i ) { g . printf( "This is Parent  
Thread: %d\n", g ). } }
```

在WIN32下，使用CreateThread函数创建线程，与Linux下创建进程不同，WIN32线程不是从创建处开始运行的，而是由CreateThread指定一个函数，线程就从那个函数处开始运行。此程序同前面的UNIX程序一样，由两个线程各打印1000条信息。threadID是子线程的线程号，另外，全局变量g是子线程与父线程共享的，这就是与Linux最大的不同之处。大家可以看出，WIN32的进程/线程要比Linux复杂，在Linux要实现类似WIN32的线程并不难，只要fork以后，让子进程调用ThreadProc函数，并且为全局变量开设共享数据区就行了，但在WIN32下就无法实现类似fork的功能了。所以

现在WIN32下的C语言编译器所提供的库函数虽然已经能兼容大多数 Linux/UNIX的库函数，但却仍无法实现fork。对于多任务系统，共享数据区是必要的，但也是一个容易引起混乱的问题，在WIN32下，一个程序员很容易忘记线程之间的数据是共享的这一情况，一个线程修改过一个变量后，另一个线程却又修改了它，结果引起程序出问题。但在Linux下，由于变量本来并不共享，而由程序员来显式地指定要共享的数据，使程序变得更清晰与安全。至于WIN32的"进程"概念，其含义则是"应用程序"，也就是相当于UNIX下的exec了。

更多优质资料尽在百考试题论坛 百考试题在线题库 linux认证
更多详细资料 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