

Linux系统下实现多线程客户_服务器 (1) 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_ti2020/628/2021_2022_Linux_E7_B3_BB_E7_BB_c103_628693.htm 在传统的Unix模型中，当一个进程需要由另一个实体执行某件事时，该进程派生（fork）一个子进程，让子进程去进行处理。Unix下的大多数网络服务器程序都是这么编写的，即父进程接受连接，派生子进程，子进程处理与客户的交互。虽然这种模型很多年来使用得很好，但是fork时有一些问题：1. fork是昂贵的。内存映像要从父进程拷贝到子进程，所有描述字要在子进程中复制等等。目前有的Unix实现使用一种叫做写时拷贝（copy - on - write）的技术，可避免父进程数据空间向子进程的拷贝。尽管有这种优化技术，fork仍然是昂贵的。2. fork子进程后，需要用进程间通信（IPC）在父子进程之间传递信息。Fork之前的信息容易传递，因为子进程从一开始就有父进程数据空间及所有描述字的拷贝。但是从子进程返回信息给父进程需要做更多的工作。线程有助于解决这两个问题。线程有时被称为轻权进程（lightweight process），因为线程比进程“轻权”，一般来说，创建一个线程要比创建一个进程快10~100倍。一个进程中的所有线程共享相同的全局内存，这使得线程很容易共享信息，但是这种简易性也带来了同步问题。一个进程中的所有线程不仅共享全局变量，而且共享：进程指令、大多数数据、打开的文件（如描述字）、信号处理程序和信号处置、当前工作目录、用户ID和组ID。但是每个线程有自己的线程ID、寄存器集合（包括程序计数器和栈指针）、栈（用于存放局部变量和返回地址）、error、信号掩码、优先级。

在Linux中线程编程符合Posix.1标准，称为Pthreads。所有的pthread函数都以pthread_开头。以下先讲述5个基本线程函数，在调用它们前均要包括pthread.h头文件。然后再给出用它们编写的一个TCP客户/服务器程序例子。第一个函数：`int pthread_create (pthread_t * tid,const pthread_attr_t * attr,void * (* func)(void *),void * arg)`；一个进程中的每个线程都由一个线程ID（thread ID）标识，其数据类型是pthread_t（常常是unsigned int）。如果新的线程创建成功，其ID将通过tid指针返回。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