

Linux系统下实现多线程客户_服务器 (2) 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_628691.htm 每个线程都有很多属性：优先级、起始栈大小、是否应该是一个守护线程等等，当创建线程时，我们可通过初始化一个pthread_attr_t变量说明这些属性以覆盖缺省值。我们通常使用缺省值，在这种情况下，我们将attr参数说明为空指针。最后，当创建一个线程时，我们要说明一个它将执行的函数。线程以调用该函数开始，然后或者显式地终止（调用pthread_exit）或者隐式地终止（让该函数返回）。函数的地址由func参数指定，该函数的调用参数是一个指针arg，如果我们需要多个调用参数，我们必须将它们打包成一个结构，然后将其地址当作唯一的参数传递给起始函数。在func和arg的声明中，func函数取一个通用指针（void *）参数，并返回一个通用指针（void *），这就使得我们可以传递一个指针（指向任何我们想要指向的东西）给线程，由线程返回一个指针（同样指向任何我们想要指向的东西）。调用成功，返回0，出错时返回正Exxx值。Pthread函数不设置errno。第二个函数：int pthread_join(pthread_t tid,void ** status).该函数等待一个线程终止。把线程和进程相比，pthread_creat类似于fork，而pthread_join类似于waitpid。我们必须等待线程的tid，很可惜，我们没有办法等待任意一个线程结束。如果status指针非空，线程的返回值（一个指向某个对象的指针）将存放在status指向的位置。第三个函数：pthread_t pthread_self(void).线程都有一个ID以在给定的进程内标识自己。线程ID由pthread_creat返回，我们可

以pthread_self取得自己的线程ID。 第四个函数： int
pthread_detach(pthread_t tid). 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