

Linux系统下实现多线程客户_服务器 (3) 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_ti2020/570/2021_2022_Linux_E7_B3_BB_E7_BB_c103_570644.htm 线程或者是可汇合的 (joinable)

或者是脱离的 (detached)。当可汇合的线程终止时，其线程ID和退出状态将保留，直到另外一个线程调用pthread_join。脱离的线程则像守护进程：当它终止时，所有的资源都释放，我们不能等待它终止。如果一个线程需要知道另一个线程什么时候终止，最好保留第二个线程的可汇合性

。Pthread_detach函数将指定的线程变为脱离的。该函数通常被想脱离自己的线程调用，如：pthread_detach(pthread_self())。第五个函数：void pthread_exit(void * status)。该函数终止线程。如果线程未脱离，其线程ID和退出状态将一直保留到调用进程中的某个其他线程调用pthread_join函数。指针status不能指向局部于调用线程的对象，因为线程终止时这些对象也消失。有两种其他方法可使线程终止：1. 启动线程的函数

(pthread_creat的第3个参数) 返回。既然该函数必须说明为返回一个void指针，该返回值便是线程的终止状态。2. 如果进程的main函数返回或者任何线程调用了exit，进程将终止，线程将随之终止。以下给出一个使用线程的TCP回射客户/服务器的例子，完成的功能是客户端使用线程给服务器发从标准输入得到的字符，并在主线程中将从服务器端返回的字符显示到标准输出，服务器端将客户端发来的数据原样返回给客户端，每一个客户在服务器上对应一个线程。利用该程序框架，通过扩展客户端和服务端端的功能，可以完成多种基于多线程的客户机/服务器程序。该程序在RedHat 6.0

和TurboLinux4.02下调试通过。 共用头文件如下：（ head.h ）

```
#include #include #include #include #include #include
#include #include #include #include #include #define
MAXLINE 1024 #define SERV_PORT 8000 #define LISTENQ
1024 static int sockfd. static FILE * fp.公用函数如下（ common.c
）：/* 从一个描述字读文本行 */ ssize_t readline(int fd, void
* vptr, size_t maxlen) { ssize_t n, rc. char c, * ptr. for (n=1. n0) { if
( (nwritten=write (fd, ptr, nleft ) )&lt;=0 ) { if (errno==EINTR )
nwritten=0. else return ( - 1). } nleft - =nwritten. ptr + + =nwritten.
} 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
```