

JAVA基础：Linux下的多线程编程 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/145/2021_2022_JAVA_E5_9F_BA_E7_A1_80_c104_145431.htm

1 引言 线程（thread）技术早在60年代就被提出，但真正应用多线程到操作系统中去，是在80年代中期，solaris是这方面的佼佼者。传统的Unix也支持线程的概念，但是在一个进程（process）中只允许有一个线程，这样多线程就意味着多进程。现在，多线程技术已经被许多操作系统所支持，包括Windows/NT，当然，也包括Linux。为什么有了进程的概念后，还要再引入线程呢？使用多线程到底有哪些好处？什么样的系统应该选用多线程？我们首先必须回答这些问题。使用多线程的理由之一是和进程相比，它是一种非常“节俭”的多任务操作方式。我们知道，在Linux系统下，启动一个新的进程必须分配给它独立的地址空间，建立众多的数据表来维护它的代码段、堆栈段和数据段，这是一种“昂贵”的多任务工作方式。而运行于一个进程中的多个线程，它们彼此之间使用相同的地址空间，共享大部分数据，启动一个线程所花费的空间远远小于启动一个进程所花费的空间，而且，线程间彼此切换所需的时间也远远小于进程间切换所需要的时间。据统计，总的说来，一个进程的开销大约是一个线程开销的30倍左右，当然，在具体的系统上，这个数据可能会有较大的区别。使用多线程的理由之二是线程间方便的通信机制。对不同进程来说，它们具有独立的数据空间，要进行数据的传递只能通过通信的方式进行，这种方式不仅费时，而且很不方便。线程则不然，由于同一进程下的线程之间共享数据空间，所以一个线程的数据

可以直接为其它线程所用，这不仅快捷，而且方便。当然，数据的共享也带来其他一些问题，有的变量不能同时被两个线程所修改，有的子程序中声明为static的数据更有可能给多线程程序带来灾难性的打击，这些正是编写多线程程序时最需要注意的地方。除了以上所说的优点外，不和进程比较，多线程程序作为一种多任务、并发的工作方式，当然有以下的优点：1) 提高应用程序响应。这对图形界面的程序尤其有意义，当一个操作耗时很长时，整个系统都会等待这个操作，此时程序不会响应键盘、鼠标、菜单的操作，而使用多线程技术，将耗时长长的操作（time consuming）置于一个新的线程，可以避免这种尴尬的情况。2) 使多CPU系统更加有效。操作系统会保证当线程数不大于CPU数目时，不同的线程运行于不同的CPU上。3) 改善程序结构。一个既长又复杂的进程可以考虑分为多个线程，成为几个独立或半独立的运行部分，这样的程序会利于理解和修改。下面我们先来尝试编写一个简单的多线程程序。

2 简单的多线程编程

Linux系统下的多线程遵循POSIX线程接口，称为pthread。编写Linux下的多线程程序，需要使用头文件pthread.h，连接时需要使用库libpthread.a。顺便说一下，Linux下pthread的实现是通过系统调用clone（）来实现的。clone（）是Linux所特有的系统调用，它的使用方式类似fork，关于clone（）的详细情况，有兴趣的读者可以去查看有关文档说明。下面我们展示一个最简单的多线程程序example1.c。

```
/* example.c */
#include <pthread.h>
#include <stdio.h>
void thread(void){int i;for(i=0;i<10;i++)printf("This is a pthread.\n");}
int main(void){pthread_t id;int i;ret;ret=pthread_create(&id,NULL,(void *)
```

```
thread,NULL).if(ret!=0){printf ("Create pthread error!\n").exit  
(1).}for(i=0.iprintf("This is the main  
process.\n").pthread_join(id,NULL).return (0).} 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
```