

平凡的程序员

喜欢编程、算法、虚拟机、搜索引擎、编译器、网络、游戏设计、操作系统等所有计算机中好玩的东西

目录视图

摘要视图

RSS 订阅

个人资料



feixiaoxing

恒

访问：1555216次
积分：19072分
排名：第111名

原创：296篇 转载：0篇
译文：0篇 评论：2085条

文章搜索

文章分类

代码测试 (4)
C/C++ (31)
数据结构和算法 (65)
多线程编程 (20)
c语言和设计模式 (26)
搜索引擎的那些事 (8)
Linux开发 (17)
操作系统 (28)
随想录 (62)
第三方库编译 (3)
linux驱动编写 (5)
verilog学习记 (4)
从skyeeye学习arm (5)
fjk学习记 (17)

文章存档

2014年06月 (3)
2014年05月 (16)
2014年04月 (2)
2014年03月 (1)

有奖征资源，博文分享有内涵 5月推荐博文汇总 大数据读书汇--获奖名单公布 2014 CSDN博文大赛

linux下的C语言开发（线程互斥）

分类：Linux开发 2012-02-08 21:39 9152人阅读 评论(6) 收藏 举报

linux 语言 c null 多线程 thread

【声明：版权所有，欢迎转载，请勿用于商业用途。联系信箱：feixiaoxing @163.com】

对于编写多线程的朋友来说，线程互斥是少不了的。在linux下面，编写多线程常用的工具其实是pthread_mutex_t。本质上来说，它和Windows下面的mutex其实是一样的，差别几乎是没有什么。希望对线程互斥进行详细了解的朋友可以看[这里](#)。

[cpp]

01. #include <stdio.h>
02. #include <pthread.h>
03. #include <unistd.h>
04. #include <stdlib.h>
05.
06. static int value = 0;
07. pthread_mutex_t mutex;
08.
09. void func(void* args)
10. {
11. while(1)
12. {
13. pthread_mutex_lock(&mutex);
14. sleep(1);
15. value ++;
16. printf("value = %d!\n", value);
17. pthread_mutex_unlock(&mutex);
18. }
19. }
20.
21. int main()
22. {
23. pthread_t pid1, pid2;
24. pthread_mutex_init(&mutex, NULL);
25.
26. if(pthread_create(&pid1, NULL, func, NULL))
27. {
28. return -1;
29. }
30.
31. if(pthread_create(&pid2, NULL, func, NULL))
32. {
33. return -1;
34. }
35.
36. while(1)
37. sleep(0);
38.
39. return 0;
40. }

编写mutex.c文件结束之后，我们就可以开始编译了。首先你需要输入gcc mutex.c -o mutex -lpthread，编译之后你就可以看到mutex可执行文件，输入./mutex即可。

2014年02月 (2) 展开

阅读排行

linux下的C语言编程（总

(50299)

一步一步写算法（之 算法

(47193)

一步一步写算法（之hasl

(43308)

用汇编的眼光看C++（之

(29959)

多线程的那点事儿（基础

(28561)

linux下的C语言开发（网

(23823)

多线程的那点事儿（之大

(23799)

一步一步写算法（之单向

(23744)

一步一步写算法（之快速

(23255)

随想录（程序员和收入）

(22290)

推荐文章

最新评论

一步一步写算法（之hash表）

Brightziu: data % 10 这个为啥要取余数呢？

多线程的那点事儿（基础篇）

EbowTang: 草，牛逼了！我就邪恶的全盘学习了！

随想录（移动app下的生活）

zistym: 为什么民营企业的app要比国营公司的app好用得多？像农业银行的app, 太烂了。国营的基本是服务领...

随想录（从apple的swift语言说走

w8708812: 博主，你好，我从去年开始拜读博主的文章了，收获很大，“我们要记住技术是为业务服务的，只有真正给客户创...

linux下的C语言开发（动态库）

AmelieDan: 简单，干练

多线程的那点事儿（之数据互斥）

springontime: 或者我认为应该写成while(flag && (turn == index))

C语言和设计模式（继承、封装、

qq584253394: 恩。其实都是主要靠指针来实现抽象了。

一步一步写算法（之A*算法）

ios-开发者: 不知道 那两个函数传入的value数组从哪里来得，想请教

用汇编的眼光看C++（之虚函数）

xueerfei: LZ这块是不是有点小问题，调用顺序应该是：edx->->[, ecx里的内容是ptr，所以是两...

一步一步写算法（之洗牌算法）

wocaonilaozi: 楼主的程序每次运行后的结果都是一样的，因为他使用的是系统默认的种子，rand()使用前要有种子；应该...

友情链接

酷壳

云风

李先静

刘未鹏

高德纳

孟岩

潘爱民

周伟明

gamers

codeproject

[cpp]

01. [test@localhost thread]\$./mutex

02. value = 1!

03. value = 2!

04. value = 3!

05. value = 4!

06. value = 5!

07. value = 6!

上一篇 linux下的C语言开发（线程等待）

下一篇 linux下的C语言开发（网络编程）

顶

4

踩

0

主题推荐

c语言

线程

linux

多线程

windows

猜你在找

一步一步写算法（之双向链表）

一步一步写算法（之二叉树广度遍历）

一步一步写算法（之爬楼梯）

linux驱动面试题汇总

linux进程创建过程与原理

htons函数详解

在centos (linux) 搭建 eclipse c++开发分环境

UNIX中后台进程与守护进程

Java之美[从菜鸟到高手演变]之字符串

【STM32F4探索套件】序列之9：UCOSII操作系统移植

免费学习IT4个月,月薪12000

中国[官方授权]IT培训与就业示范基地,学成后名企直接招聘,月薪12000起!

查看评论

4楼 richardlxc007 2012-09-26 16:33发表

C

我在linux下运行，把void fun(void *args) 修改为 void *fun(void *args)，才能够编译运行成功。

3楼 Dragon_L 2012-05-22 21:49发表

个人的浅显理解，不知道解释的是否合适。。。

2楼 yuefenghuang 2012-03-02 16:33发表

C

sleep(0) 是什么意思呢？不休眠一直while(1)，不就是忙式等待嘛？

Re: Dragon_L 2012-05-22 21:48发表

回复yuefenghuang：创建两个线程对应的线程处理函数都是void fun(void *args); 是为了两个线程都能对value进行加1操作吧。。。 main函数中while条件一直为真，sleep(0); 是一直主线程不结束，等待着两个线程对value轮流加1并打印输出吧。。。

Re: Dragon_L 2012-05-22 21:43发表

回复yuefenghuang：

1楼 yesman001 2012-02-23 14:31发表

C

good basic

您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

http://blog.csdn.net/feixiaoxing/article/details/7243664

2/3

华清远见
51cto
IT 人生
raw-os

核心技术类目

全部主题 [Java](#) [VPN](#) [Android](#) [iOS](#) [ERP](#) [IE10](#) [Eclipse](#) [CRM](#) [JavaScript](#) [Ubuntu](#) [NFC](#)
[WAP](#) [jQuery](#) [数据库](#) [BI](#) [HTML5](#) [Spring](#) [Apache](#) [Hadoop](#) [.NET](#) [API](#) [HTML](#) [SDK](#)
[IIS](#) [Fedora](#) [XML](#) [LBS](#) [Unity](#) [Splashtop](#) [UML](#) [components](#) [Windows Mobile](#) [Rails](#)
[QEMU](#) [KDE](#) [Cassandra](#) [CloudStack](#) [FTC](#) [coremail](#) [OPhone](#) [CouchBase](#) [云计算](#) [iOS6](#)
[Rackspace](#) [Web App](#) [SpringSide](#) [Maemo](#) [Compuware](#) [大数据](#) [aptech](#) [Perl](#) [Tornado](#) [Ruby](#)
[Hibernate](#) [ThinkPHP](#) [Spark](#) [HBase](#) [Pure](#) [Solr](#) [Angular](#) [Cloud Foundry](#) [Redis](#) [Scala](#)
[Django](#) [Bootstrap](#)

公司简介 | 招贤纳士 | 广告服务 | 银行汇款帐号 | 联系方式 | 版权声明 | 法律顾问 | 问题报告 | 合作伙伴 | 论坛反馈

网站客服 杂志客服 微博客服 webmaster@csdn.net 400-600-2320

京 ICP 证 070598 号

北京创新乐知信息技术有限公司 版权所有

江苏乐知网络技术有限公司 提供商务支持

Copyright © 1999-2014, CSDN.NET, All Rights Reserved 