

原

android NDK编程：使用posix多线程与mutex互斥同步

2014年09月28日 17:28:57   hai836045106   阅读数：6342   [更多](#)

MainActivity.java

调用原生方法   `posixThreads(int threads, int iterations)`   启动线程

```
1 package com.apress.threads;
2
3 import android.app.Activity;
4 import android.os.Bundle;
5 import android.view.View;
6 import android.view.View.OnClickListener;
7 import android.widget.Button;
8 import android.widget.EditText;
9 import android.widget.TextView;
10
11 public class MainActivity extends Activity {
12
13     private EditText editThreads;
14     private EditText editIterations;
15     private Button btnStart;
16     private TextView tvLog;
17
18     @Override
19     protected void onCreate(Bundle savedInstanceState) {
20         super.onCreate(savedInstanceState);
21         setContentView(R.layout.activity_main);
22
23         nativeInit();
24         editThreads = (EditText) findViewById(R.id.threads_edit);
25         editIterations = (EditText) findViewById(R.id.iterations_edit);
26         btnStart = (Button) findViewById(R.id.start_button);
27         tvLog = (TextView) findViewById(R.id.log_view);
28
29         btnStart.setOnClickListener(new OnClickListener() {
30
31             @Override
32             public void onClick(View v) {
33
```



```
36 | 37 |         if (threads > 0 && iterations > 0) {
38 |             startThreads(threads, iterations);
39 |         }
40 |     }
41 | });
42 | }
43 |
44 | private void startThreads(int threads, int iterations) {
45 |     // javaThreads(threads, iterations); // 使用java的线程来循环
46 |     posixThreads(threads, iterations); // 使用posix线程
47 | }
48 |
49 | private void javaThreads(int threads, final int iterations) {
50 |     for (int i = 0; i < threads; i++) {
51 |         final int id = i;
52 |         new Thread() {
53 |
54 |             @Override
55 |             public void run() {
56 |                 nativeWorker(id, iterations);
57 |                 super.run();
58 |             }
59 |
60 |         }.start();
61 |     }
62 | }
63 |
64 | private void onNativeMessage(final String message) {
65 |     runOnUiThread(new Runnable() {
66 |
67 |         @Override
68 |         public void run() {
69 |             tvLog.append(message);
70 |             tvLog.append("\n");
71 |         }
72 |     });
73 | }
74 |
75 | private native void posixThreads(int threads, int iterations);
76 |
77 | // 初始化
78 | private native void nativeInit();
79 |
80 | // 释放内存
81 | private native void nativeFree();
82 |
83 | // java线程直接调用jni
84 | ..
```

```
87 |  
88 |         int value;  
89 |         try {  
90 |             value = Integer.parseInt(editText.getText().toString());  
91 |         } catch (Exception e) {  
92 |             value = defaultValue;  
93 |         }  
94 |         return value;  
95 |     }  
96 |  
97 |     @Override  
98 |     protected void onDestroy() {  
99 |         nativeFree();  
100 |         super.onDestroy();  
101 |     }  
102 |  
103 |     static {  
104 |         System.loadLibrary("Threads");  
105 |     }  
106 |  
107 | }
```



com\_apress\_threads\_MainActivity.h:

```
1  /* DO NOT EDIT THIS FILE - it is machine generated */  
2  #include <jni.h>  
3  /* Header for class com_apress_threads_MainActivity */  
4  
5  #ifndef _Included_com_apress_threads_MainActivity  
6  #define _Included_com_apress_threads_MainActivity  
7  #ifdef __cplusplus  
8  extern "C" {  
9  #endif  
10 /*  
11  * Class:      com_apress_threads_MainActivity  
12  * Method:     posixThreads  
13  * Signature:  (II)V  
14  */  
15 JNIEXPORT void JNICALL Java_com_apress_threads_MainActivity_posixThreads  
16     (JNIEnv *, jobject, jint, jint);  
17  
18 /*  
19  * Class:      com_apress_threads_MainActivity
```

```
22 | */23 | JNIEXPORT void JNICALL Java_com_apress_threads_MainActivity_nativeInit
24 | (JNIEnv *, jobject);
25 |
26 | /*
27 | * Class:      com_apress_threads_MainActivity
28 | * Method:     nativeFree
29 | * Signature: ()V
30 | */
31 | JNIEXPORT void JNICALL Java_com_apress_threads_MainActivity_nativeFree
32 | (JNIEnv *, jobject);
33 |
34 | /*
35 | * Class:      com_apress_threads_MainActivity
36 | * Method:     nativeWorker
37 | * Signature: (II)V
38 | */
39 | JNIEXPORT void JNICALL Java_com_apress_threads_MainActivity_nativeWorker
40 | (JNIEnv *, jobject, jint, jint);
41 |
42 | #ifdef __cplusplus
43 | }
44 | #endif
45 | #endif
```



com\_apress\_threads\_MainActivity.cpp:

在java层调用到原生方法Java\_com\_apress\_threads\_MainActivity\_posixThreads后，封装参数，调用pthread\_create创建线程，使用pthread\_join监听线程运行结果并回调到java层。

线程指向的函数为**void\*** nativeWorkerThread(**void\*** args),在这个函数里将native线程通过JNI来attach到Java环境里，这样native线程才可以使用JNIEnv，执行完毕后要Detach。具体说明可参考[http://docs.oracle.com/javase/6/docs/technotes/guides/jni/spec/invoke.html#attach\\_current\\_thread](http://docs.oracle.com/javase/6/docs/technotes/guides/jni/spec/invoke.html#attach_current_thread)

native线程调用nativeWorker函数输出字符串，通过JNIEnv回调java方法。

在初始化方法中初始化JavaVM\* gVm，用来attach到虚拟机中。

```
1 | #include <stdio.h>
2 | #include <unistd.h>
3 | #include <pthread.h>
4 |
5 | #include "com_apress_threads_MainActivity.h"
6 | #include <android/log.h>
7 |
8 | #define LOG_TAG "LOG FROM INT"
```

2019人工智能薪资

Python资料免费领

会员任意学

Java薪资多少

怎样才能不被裁员

永久加速器

知网论文查重入口

登录

注册

×

```
11 | 12 | //传递pthread参数用的结构体
13 | struct NativeWorkerArgs {
14 |     jint id;
15 |     jint iterations;
16 | };
17 |
18 | //回调java的方法
19 | static jmethodID gOnNativeMessage = NULL;
20 | static JavaVM* gVm = NULL; //虚拟机引用,作为全局变量
21 | static jobject gObj = NULL;
22 | static pthread_mutex_t mutex;
23 |
24 | //loadLibrary的时候自动调用,在这里获得全局vm引用
25 | jint JNI_OnLoad(JavaVM* vm, void* reserved) {
26 |     gVm = vm;
27 |
28 |     LOGW("JNI_OnLoad");
29 |     return JNI_VERSION_1_4;
30 | }
31 |
32 | void Java_com_apress_threads_MainActivity_nativeInit(JNIEnv *env, jobject obj) {
33 |
34 |     //初始化互斥量
35 |     if (0 != pthread_mutex_init(&mutex, NULL)) {
36 |         //异常
37 |         jclass exceptionClazz = env->FindClass("java/lang/RuntimeException");
38 |         //抛出
39 |         env->ThrowNew(exceptionClazz, "Unable to init mutex--");
40 |     }
41 |
42 |     if (NULL == gObj) {
43 |         gObj = env->NewGlobalRef(obj);
44 |     }
45 |
46 |     //初始java回调
47 |     if (NULL == gOnNativeMessage) {
48 |         jclass clazz = env->GetObjectClass(obj);
49 |         gOnNativeMessage = env->GetMethodID(clazz, "onNativeMessage",
50 |             "(Ljava/lang/String;)V");
51 |
52 |         if (NULL == gOnNativeMessage) {
53 |             //异常
54 |             jclass exceptionClazz = env->FindClass(
55 |                 "java/lang/RuntimeException");
56 |             //抛出
57 |             env->ThrowNew(exceptionClazz, "Unable to find method--");
58 |         }
59 |     }
```

```
62
63 void Java_com_apress_threads_MainActivity_nativeFree(JNIEnv *env, jobject) {
64
65     // 释放全局变量
66     if (NULL != gObj) {
67         env->DeleteGlobalRef(gObj);
68         gObj = NULL;
69     }
70
71     // 释放互斥量
72     if (0 != pthread_mutex_destroy(&mutex)) {
73         // 异常
74         jclass exceptionClazz = env->FindClass("java/lang/RuntimeException");
75         // 抛出
76         env->ThrowNew(exceptionClazz, "Unable to destroy mutex--");
77     }
78 }
79
80 // ndk线程执行的代码
81 void nativeWorker(JNIEnv *env, jobject obj, jint id, jint iterations) {
82
83     // lock
84     if (0 != pthread_mutex_lock(&mutex)) {
85         // 异常
86         jclass exceptionClazz = env->FindClass("java/lang/RuntimeException");
87         // 抛出
88         env->ThrowNew(exceptionClazz, "Unable to lock mutex--");
89         return;
90     }
91
92     for (jint i = 0; i < iterations; i++) {
93         char message[26];
94         sprintf(message, "Worker %d:Iteration %d", id, i);
95
96         // 回调java方法
97         jstring messageString = env->NewStringUTF(message);
98         env->CallVoidMethod(obj, gOnNativeMessage, messageString);
99
100         if (NULL != env->ExceptionOccurred()) {
101             break;
102         }
103         sleep(1);
104     }
105
106     // unlock
107     if (0 != pthread_mutex_unlock(&mutex)) {
108         // 异常
109         jclass exceptionClazz = env->FindClass("java/lang/RuntimeException");
```

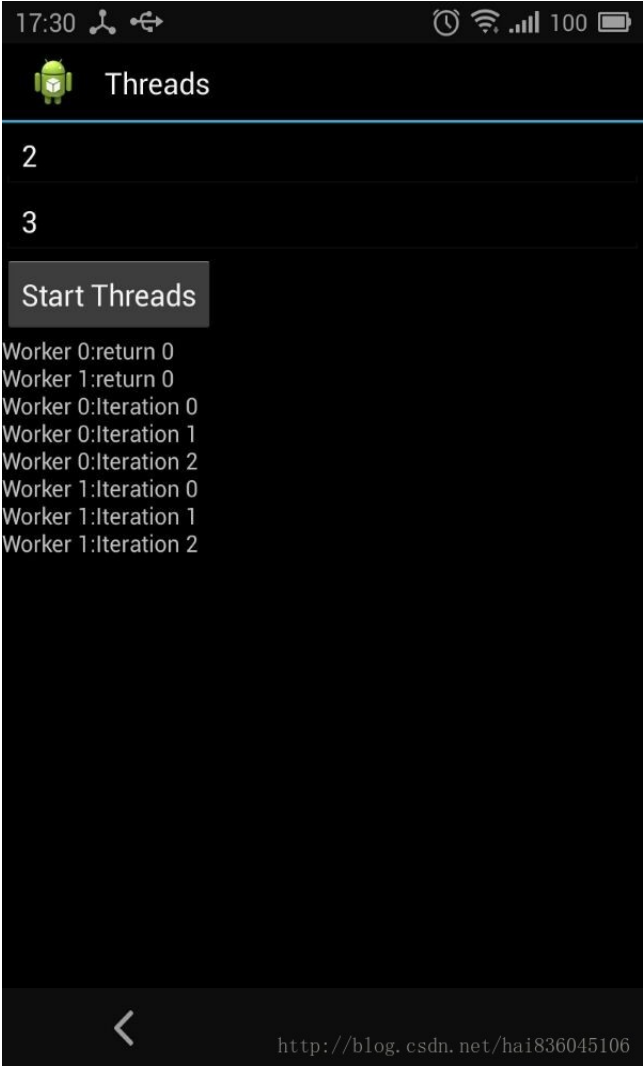


```
112 | 113 |      }
114 | }
115 |
116 | void Java_com_apress_threads_MainActivity_nativeWorker(JNIEnv *env, jobject obj,
117 |     jint id, jint iterations) {
118 |     nativeWorker(env, obj, id, iterations);
119 | }
120 |
121 | //pthread执行的方法
122 | static void* nativeWorkerThread(void* args) {
123 |     JNIEnv* env = NULL;
124 |     if (0 == gVm->AttachCurrentThread(&env, NULL)) {
125 |         NativeWorkerArgs* nativeWorkerAgrs = (NativeWorkerArgs*) args;
126 |
127 |         //
128 |         nativeWorker(env, gObj, nativeWorkerAgrs->id,
129 |             nativeWorkerAgrs->iterations);
130 |
131 |         delete nativeWorkerAgrs;
132 |
133 |         gVm->DetachCurrentThread();
134 |     }
135 |
136 |     return (void*) 1;
137 | }
138 |
139 | //java调用的,启动多个线程
140 | void Java_com_apress_threads_MainActivity_posixThreads(JNIEnv *env, jobject obj,
141 |     jint threads, jint iterations) {
142 |
143 |     //thread handlers
144 |     pthread_t* handles = new pthread_t[threads];
145 |
146 |     //启动线程
147 |     for (jint i = 0; i < threads; i++) {
148 |
149 |         //thread arguments
150 |         NativeWorkerArgs* nativeWorkArgs = new NativeWorkerArgs();
151 |         nativeWorkArgs->id = i;
152 |         nativeWorkArgs->iterations = iterations;
153 |
154 |         //thread handler
155 |         int result = pthread_create(&handles[i], NULL, nativeWorkerThread,
156 |             (void*) nativeWorkArgs);
157 |
158 |         if (result != 0) {
159 |             //异常
```

```
162         //抛出163|         env->ThrowNew(exceptionClazz, "Unable to create thread--");
164         return;
165     }
166 }
167
168 // 线程运行结果
169 for (jint i = 0; i < threads; i++) {
170     void** result = NULL;
171     if (0 != pthread_join(handles[i], result)) {
172         //异常
173         jclass exceptionClazz = env->FindClass(
174             "java/lang/RuntimeException");
175         //抛出
176         env->ThrowNew(exceptionClazz, "Unable to join thread--");
177     } else {
178         char message[26];
179         sprintf(message, "Worker %d:return %d", i, result);
180
181         jstring messageString = env->NewStringUTF(message);
182         env->CallVoidMethod(obj, gOnNativeMessage, messageString);
183
184         if (NULL != env->ExceptionOccurred()) {
185             return;
186         }
187     }
188 }
189
190 }
```

这里执行的时候会阻塞界面，直接所有native线程运行完毕。





代码下载: <http://download.csdn.net/detail/hai836045106/7986143>

Java培训需要多少钱？

零基础学IT选Java，易学、高薪、前景广，100万人才缺口，互联网必备人才

相对作者说占什么

2019人工智能薪资

Python资料免费领

会员任意学

Java薪资多少

怎样才能不被裁员

永久加速器

知网论文查重入口

登录

注册

×

jni和c/c++多线程的处理

5041

java中要访问C++代码时, 使用JNI是唯一选择. 然而,在多线程的情况下, 可能出现以下问题:问题描述:一个java对象通...

来自：1984的面包机

Android NDK线程篇（四）

1515

本篇重点介绍原生线程的相关知识，包括线程基础知识，线程同步等，学完本篇，你将会对NDK原生线程有一个更...

来自：kgdwbb的专栏

android底层库libutils之Mutex,Autolock研究

714

1)认识Mutex,Autolock Mutex其实是对linux 线程互斥锁做了一层封装，由下面的类定义中可以发现pthread\_mutex\_t...

来自：armwind的专栏

超系统的Java学习路线规划，你一定不知道的薪资翻倍秘诀！

百万程序员点赞的学习路线，你看过吗？

Android同步类：Mutex和Condition的使用方式

1144

Metux类Mutex是互斥类，多个线程访问同一个资源时，要确保他们的互斥。 Autolock类是Mutex的一个内部类，方...

来自：Manso的空间

android mutex 详细介绍

489

一个进程中不可能只有一个线程在战斗，所以一个进程中一般都是有多个线程的同时协助工作，多线程情况下，对...

来自：xiongtiancheng的博客

Android中 Mutex的理解

3262

Android libutils 中定义了 Mutext 和 Mutex::AutoLock 对象来进行临界资源的加锁和解锁。Mutex：：AutoLock 在构...

来自：lsdmx2016的总结

Android中线程同步之Mutex与Condtion的用法

1.3万

田海立2013-08-10Android封装的同步类主要有MUTEX（AutoMutex）与Condition。本文分析如何使用他们，具体...

来自：田海立@CSDN

jni 线程退出时的问题

07-04

native thread exited without detaching 我调用的别人的.a文件 他们设置的虚函数，我实现方法 流程是 我调用一个函数 他们在a文件中开线程 然后回调虚...

用它兑水喝！六旬老汉都能坚持一小时不射！何况你！

远中宏建·熾燚

Android studio日常bug总结

294

1、 Cannot find System Java Compiler. Ensure that you have installed a JDK (not just a JRE) and config...

来自：聪哥的专栏

文章热词

编程语言学哪个好 Android Android培训 Android课程 Android学习

相关热词

c#多线程mutex c++使用mutex c++ 同一个变量互斥 多线程 c++ 变量 同步 多线程 c++ mutex 使用 python编程编程培训班 编程python培训

Android studio ndk 编译

420

本文详细介绍了Android Studio中NDK编译的步骤和常见问题，包括环境配置、编译选项设置、以及解决常见的编译错误...

2019人工智能薪资

Python资料免费领

会员任意学

Java薪资多少

怎样才能不被裁员

永久加速器

知网论文查重入口

登录

注册

×

https://blog.csdn.net/hai836045106/article/details/39644215

10/17

<div></div> <div>218篇文章 排名:829</div> <div>关注</div>	<div></div> <div>381篇文章 排名:399</div> <div>关注</div>	<div></div> <div>187篇文章 排名:4000+</div> <div>关注</div>	<div></div> <div>171篇文章 排名:1000+</div> <div>关注</div>
<div><div>Android NDK使用之--用pthread实现多线程</div><div>Android系统本身包括Java和Linux内核；因此在Android应用中实现多线程就可以是基于Java上的和基于Linux上的...</div><div>来自：jike0901xuye的专栏</div></div>			
<div><div>Android NDK c创建新的线程</div><div>在jni的c/c++层创建一个新的线程只需要3步：1.导入库 #include 2.写好线程要做的事 void* run_1(void*); void* run_...</div><div>来自：q291611265的专栏</div></div>			
<div><div>Android多线程研究（5）——线程之间共享数据</div><div>一、如果是每个线程都执行相同的代码，则可以使用同一个Runnable来实现共享public class MultiThreadShareDat...</div><div>来自：水寒</div></div>			
<div><div>用它兑水喝！六旬老汉都能坚持一小时不射！何况你！</div><div>新创嘉意·熾燚</div></div>			
<div><div>Android之多线程实现方式及并发与同步</div><div>前言：说到多线程，就不得不先说它和进程的关系，这里先简单解释一下，当一个app程序启动时系统默认有一个进...</div><div>来自：嘻哈包袱铺 专栏</div></div>			
<div><div>android不同线程之间数据交互</div><div>跨线程(不是跨进程，一个程序间主要是跨线程，不同程序间跨进程，当然也跨线程)的数据交互，我感觉这个问题...</div><div>来自：d_o_n_g2的博客</div></div>			
<div><div>Android NDK C++ 实现生产者消费者模式（线程加减锁）</div><div>Android NDK C++ 实现生产者消费者模式（线程加减锁） 前言 生产者消费者模式是什么？可以自己百度查一下，...</div><div>来自：yangyasong的博客</div></div>			
<div><div>使用NDK原生线程处理多线程问题-互斥锁的使用</div><div>前言：NDK编程既可以用C++也可以用C。但是笔者在NDK开发过程中涉及到多线程的问题，发现直接套用C++的...</div><div>来自：gezigezao的博客</div></div>			
<div><div>Android中同步类Mutex（AutoMutex）与Condition。</div><div>在Android中，封装的同步类主要有Mutex（AutoMutex）与Condition。这两个类在android中被大量的使用，这也...</div><div>来自：andy_android的专栏</div></div>			
<div><div>老中医说：饭后用一物，体重不过百！北京人必看！</div><div>楠洲·熾燚</div></div>			
<div><div>Android JNI开发（6）--JNI中使用线程</div><div>JNI中线程是基于Linux的线程，包含头文件 #includepthread_mutex_t mutex;互斥锁 pthread_cond_t cond;条件...</div><div>来自：猿诗人的博客</div></div>			
<div><div>JNI native多线程调用Java静态方法</div><div>前言：最近在做音视频实时通信，因为回声问题要解决，而播放和音视频采集是异步处理，所以需要java在中间做...</div><div>来自：hongdameng的专栏</div></div>			
<div><div>如何在多线程中使用JNI？</div><div></div></div>			
2019人工智能薪资	Python资料免费领	会员任意学	Java薪资多少
怎样才能不被裁员	永久加速器	知网论文查重入口	
登录	注册	×	

<div><div><div>NDK 5 使用POSIX线程</div><div>Java中使用线程特别简单，实现Runnable接口 或者 继承Thread. NDK中使用线程可以在Activity中直接调用java线...</div></div><div><div>3860</div><div>来自： <a href="#">移动生活，幸福代码</a></div></div></div>
<div><div><div>POSIX线程互斥锁</div><div>一、什么是互斥锁      另一种在多线程程序中同步访问手段是使用互斥量。程序员给某个对象加上一把“锁”，每次...</div></div><div><div>593</div><div>来自： <a href="#">ycc541的专栏</a></div></div></div>
<div><div><div>千万不要再乱喝蜂蜜了，美女亲赴深山，揭露背后惊人黑幕！</div><div>聚优 · 顶新</div></div></div>
<div><div><div>Linux多线程实践(5) --Posix信号量与互斥量解决生产者消费者问题</div><div>Posix信号量Posix 信号量有名信号量无名信号量sem_opensem_initsem_closesem_destroysem_unlink sem_waits...</div></div><div><div>3299</div><div>来自： <a href="#">菜鸟-翡青</a></div></div></div>
<div><div><div>【VS2010】C++多线程同步与互斥简单运用</div><div>转自：http://blog.csdn.net/ccing/article/details/6215998 继以往的想法，写这点文字，粘贴点代码，是为了增加自己...</div></div><div><div>3204</div><div>来自： <a href="#">世事洞明皆学问</a></div></div></div>
<div><div><div>C#多线程开发8：线程同步之Mutex类</div><div>以下代码模拟“从自动取款机中取钱”的操作：当账户余额大于等于1000时，取出1000放入口袋。代码中使用两个线...</div></div><div><div>2813</div><div>来自： <a href="#">三五月儿的专栏</a></div></div></div>
<div><div><div>C++11 多线程同步 互斥锁 条件变量</div><div>在多线程程序中，线程同步（多个线程访问一个资源保证顺序）是一个非常重要的问题，Linux下常见的线程同步的...</div></div><div><div>4947</div><div>来自： <a href="#">杨博东的博客</a></div></div></div>
<div><div><div>JNI多个线程中使用</div><div>上一篇文章说到 JNIEnv 是一个与线程相关的变量，即线程A有一个 JNIEnv变量， 线程B也有一个JNIEnv变量，由...</div></div><div><div>1.1万</div><div>来自： <a href="#">andyhuabing的专栏</a></div></div></div>
<div><div><div>千万不要再乱喝蜂蜜了!美女亲赴深山，发现惊人真相！</div><div>聚优 · 顶新</div></div></div>
<div><div><div>JNI线程</div><div>JNI多个本地线程进入Java层,Java层线程进入C本地函数，线程安全</div></div><div><div>5039</div><div>来自： <a href="#">shaohuazuo的专栏</a></div></div></div>
<div><div><div>jni中的线程, 同步以及生产者消费者模式</div><div>编译环境centos6.x64 线程创建以及结束 连接和分离线程 例子1 例子2 多线程问题 线程的同步 生产者消费者 阻塞...</div></div><div><div>58</div><div>来自： <a href="#">六道对穿肠的学习之旅</a></div></div></div>
<div><div><div>Android NDK双进程守护(Socket)</div><div>NDK双进程守护（单工机制）最近在系统的学习Android NDK开发于是想着把刚学完的一个知识点总结写一篇博客...</div></div><div><div>230</div><div>来自： <a href="#">云桥</a></div></div></div>
<div><div><div>Android实现双进程守护</div><div>如何保证Service不被Kill有关Service的知识请参考Android Service全面解析这篇文章，写的很详细。（1）onStartC...</div></div><div><div>8347</div><div>来自： <a href="#">Watson的博客</a></div></div></div>
<div><div><div>JNI 的多线程</div><div></div></div><div><div>1 4 万</div><div></div></div></div>

<div>北京蜂蜜市场黑幕惊人！知情人士冒死赴深山，揭露背后真相</div> <div>邱拉 · 顶新</div>		
<div>Android 中 C++ Thread线程用法</div> <div>Android对Linux线程提供了C++封装Thread类，它是线程的基类。使用Thread类，需创建一个新类继承于Thread类...</div>	<div>4486</div> <div>来自：<a href="#">unbroken</a></div>	
<div>NDK学习( 三 ) ，多线程与异步回调</div> <div>场景：在Java中传递任意类型参数，在JNI中新开子线程，将传入的参数处理后回调Java中的方法 技术点： 1、在J...</div>	<div>1184</div> <div>来自：<a href="#">mxw3755的专栏</a></div>	
<div>Android NDK 链接posix线程库</div> <div>posix pthread通常链接pthread在CMake脚本中都是这样做的 target_link_libraries(xxx pthread)然而在Android NDK...</div>	<div>205</div> <div>来自：<a href="#">sliver.chen的博客</a></div>	
<div>window下用android-ndk-r9d编译.so文件之mutex in namespace 'std' does not name a type</div> <div>前段时间为了封装了cocos2dx游戏代码，就编译了2个平台下的开发库，window下的.lib和.dll，Android下的.a和.so...</div>	<div>5292</div> <div>来自：<a href="#">msdb198901的博客</a></div>	
<div>多线程并发编程（四）：多线程同步互斥Wait/Notify</div> <div>前言 前面说了使用Synchronized来进行线程之间的同步，接下来说明wait/notify的使用。 首先wait/notify必须结合sy...</div>	<div>1584</div> <div>来自：<a href="#">Saytime</a></div>	
<div>C++多线程实例之互斥量同步</div> <div>要点：将互斥量 + 条件变量/事件对象封装为了一个类CMutexLock,该类同时支持windows和linux下的互斥量同步。 ...</div>	<div>4085</div> <div>来自：<a href="#">自由理想的足迹</a></div>	
<div>python 线程的同步和互斥</div> <div>先从一个简单的demo开始说起：</div>	<div>8406</div> <div>来自：<a href="#">写代码的小孩</a></div>	
<div>C++多线程同步之Mutex(互斥量)</div> <div>一、互斥量Mutex同步多线程1、Win32平台相关函数和头文件#include HANDLE CreateMutex( LPSECURITY_ATT...</div>	<div>3448</div> <div>来自：<a href="#">~探路者~的博客</a></div>	
<div>linux线程互斥与同步（part1）—互斥锁（mutex）的原理及其实现机制</div> <div>一段代码引发的问题首先，我们来编写一段代码，它的目的是定义一个全局变量，创建两个线程对其进行5000++的...</div>	<div>1874</div> <div>来自：<a href="#">My Coding Life</a></div>	
<div>Mutex::AutoLock介绍</div> <div>互斥类—Mutex Mutex是互斥类，用于多线程访问同一个资源的时候，保证一次只有一个线程能访问该资源。在《W...</div>	<div>1.8万</div> <div>来自：<a href="#">southcamel的专栏</a></div>	
<div>Android中的同步与Mutex</div>	<div>6324</div>	

<b>Android中三种锁的用法</b>		👁 9109
一、synchronized 略 二、lock Lock是java.util.concurrent.locks包下的接口，Lock 实现提供了比使用synchronized...		来自： <a href="#">可乐</a>
<b>android 里面的 AutoMutex</b>		👁 3669
在 frameworks/base/libs/binder/ProcessState.cpp 的 startThreadPool() 函数中， 函数一开始，就声明了： AutoMu...		来自： <a href="#">batoom的专栏</a>
<b>Android多线程研究（3）——线程同步和互斥及死锁</b>		👁 4279
为什么会有线程同步的概念呢？为什么要同步？什么是线程同步？先看一段代码：package com.maso.test; public c...		来自： <a href="#">水寒</a>
<b>Android 使用NDK编写 基于C层的守护进程</b>		👁 4372
做过android开发的人应该都知道GC会在资源不够用的时候会无情的回收掉我们写的进程，但是有时候我们需要我...		来自： <a href="#">Leon的小黑屋</a>
		
<b>在ndk中尝试使用原生线程</b>		👁 888
先看执行结果 第一个参数是要创建的线程数,第二个参数是每个线程执行的任务数 ndk编译结果 编译的时候可以用 ...		来自： <a href="#">hb707934728的博客</a>
 <b>Android App开发入门：使用Android Studio 2.X开发环境</b>		
本书从初学者的角度出发，通过通俗易懂的语言、丰富的实例，详细介绍使用AndroidStudio2.X开发环...		
<b>很黄很暴力的十个网站</b>		👁 74924
13岁的北京学生张某，在去年12月27日19时新闻联播一则关于净化网络视听的新闻里，接受采访时说的话激起了轩...		来自： <a href="#">Kinb_huangwei的专栏</a>
<b>史上最简单的 SpringCloud 教程   终章</b>		👁 1249397
转载请标明出处： http://blog.csdn.net/forezp/article/details/70148833 本文出自方志朋的博客 错过了这一篇，你可...		来自： <a href="#">方志朋的专栏</a>
<b>webstorm 2018 激活破解方法大全</b>		👁 692660
webstorm 作为最近最火的前端开发工具,也确实对得起那个价格,但是秉着勤俭节约的传统美德,我们肯定是能省则省...		来自： <a href="#">唐大帅的编程之路</a>
<b>初次简单逆向破解安卓APP</b>		👁 5748
最近下载了一个直播app，因为有会员时间限制,只能只用十分钟，但最近接触了一些逆向知识，抱着尝试的态度尝...		来自： <a href="#">无聊看风景的博客</a>
<b>Kotlin实战(一)</b>		👁 2470
Kotlin实战(一) 1、简介 Kotlin 是一个用于现代多平台应用的静态编程语言，由 JetBrains 开发。 Kotlin可以编译成Ja...		来自： <a href="#">zping0808的博客</a>
<b>Webstorm 最新激活码 多种破解方式(持续更新...)</b>		👁 87382
方法：License server 注册 安装完成，打开Webstorm，在弹出的License Activation窗口中选择“License server”，...		来自： <a href="#">老妖儿的博客</a>

<div>2018最新Web前端经典面试题及答案</div> <div>本篇收录了一些面试中经常会遇到的经典面试题以及自己面试过程中遇到的一些问题，并且都给出了我在网上收集...</div>	<div>👁 363361</div> <div>来自： <a href="#">wdlhao的博客</a></div>
<div>最新迅雷“应版权方要求，文件无法下载”的解决办法</div> <div>迅雷下载有的电影电视剧的时候会出现：应版权方要求，文件无法下载，或者显示迅雷任务包含违规内容 无法继续...</div>	<div>👁 231127</div> <div>来自： <a href="#">徐奕的专栏</a></div>
<div>pyCharm最新2018激活码</div> <div>本教程对jetbrains全系列可用例：IDEA、WebStorm、phpstorm、clion等 因公司的需求，需要做一个爬取最近上映...</div>	<div>👁 1270323</div> <div>来自： <a href="#">昌昌</a></div>
<div>webgl第26课-在图形上贴图片</div> <div>需要电子档书籍可以Q群：828202939 希望可以和大家一起学习、一起进步！！上一节课我们学习了 颜色与纹理...</div>	<div>👁 1911</div> <div>来自： <a href="#">谷子的博客</a></div>
<div>整理了10个干净、好用的BT、磁力链搜索网站给大家</div> <div>现在越来越流行在线看视频了，但是对于我得收藏癖爱好者，还是希望可以有比较好的资源网站的，尤其是种子、...</div>	<div>👁 107685</div> <div>来自： <a href="#">YXAPP的技术分享</a></div>
<div>颈椎病的症状,Pyhton结合治疗仪让治疗与众不同</div> <div>第1部分 Python内建对象 第1章 Python对象初探 1.1 Python内的对象 1.1.1 对象机制的基石——PyObject 1.1.2 定...</div>	<div>👁 7994</div> <div>来自： <a href="#">john_dung的博客</a></div>
<div>webgl第30课-用键盘来改变视点</div> <div>需要电子档书籍可以Q群：828202939 希望可以和大家一起学习、一起进步！！所有的课程源代码在我上传的资...</div>	<div>👁 1910</div> <div>来自： <a href="#">谷子的博客</a></div>
<div>webstorm 2018破解版本【mac适用】下载（解压直接可用）</div> <div>webstorm激活码据内部人员透露封查太严，是故分享一套三步即可破解方法：【权威】官网最新版本破解永久有效...</div>	<div>👁 10109</div> <div>来自： <a href="#">赏樱看雪撸代码</a></div>
<div>微服务Springcloud超详细教程+实战（十）</div> <div>本人正在找深圳Java实习工作，求大佬带飞 QQ：1172796094 如在文档中遇到什么问题请联系作者 —————...</div>	<div>👁 14066</div>
<div>微服务Springcloud超详细教程+实战（八）</div> <div>如在文档中遇到什么问题请联系作者 QQ：1172796094 本人正在找深圳Java实习工作，求大佬带飞 —————...</div>	<div>👁 17274</div>
<div>史上最全Java面试题（带全部答案）</div> <div>今天要谈的主题是关于求职，求职是在每个技术人员生涯中都要经历多次。对于我们大部分人而言，在进入自己...</div>	<div>👁 126904</div> <div>来自： <a href="#">林老师带你学编程</a></div>
<div>【Android学习】第三章·存储容量的获取&amp;xml格式文本的创建与解析</div> <div>相对布局：结合RelativeLayout九宫格 表格布局和绝对布局不常用，了解就好 谷歌替代system.out.println()用Lo...</div>	<div>👁 1555</div> <div>来自： <a href="#">唐三十胖子的博客</a></div>
<div>2018最好用百度云破解版，百度网盘不限速下载，教你如何解决百度网盘限速的方法。亲测完美使用</div> <div>百度网盘不限速 点击下载 提取码：jsk0 百度网盘不限速 点击下载 提取码：jsk0 对于大多数人来说，每次在百度网...</div>	<div>👁 98216</div> <div>来自： <a href="#">qq_41925894的博客</a></div>
<div>Proxyee-down的下载与安装教程</div>	<div>👁 146025</div>

<div>Webstorm注册码 永久有效 欢迎大家使用</div> <div>2RRJMBXW33-eyJsaWNlbnNISWQiOilyUIJKTUJYVzMzliwibGljZW5zZWVOYW1lIjoi5b285bK4IHNVZnR3YXJlMil...</div>	<div>9301</div> <div>来自： <a href="#">GitHuang</a></div>
<div>webgl第27课-三维空间-不同位置的三角形</div> <div>需要电子书籍可以Q群：828202939 希望可以和大家一起学习、一起进步！！上一节课我们学习了 颜色与纹理...</div>	<div>1973</div> <div>来自： <a href="#">谷子的博客</a></div>
<div>webgl第28课-三维空间之加入旋转矩阵的三角形</div> <div>需要电子书籍可以Q群：828202939 希望可以和大家一起学习、一起进步！！上一节课我们学习了 三维空间-...</div>	<div>1968</div> <div>来自： <a href="#">谷子的博客</a></div>
<div>微服务Springcloud超详细教程+实战（九）</div> <div>如在文档中遇到什么问题请联系作者 QQ：1172796094 本人正在找深圳Java实习工作，求大佬带飞—————...</div>	<div>14354</div> <div></div>
<div>军事理论课答案（西安交大版）</div> <div>1.1 1 【单选题】我国陆地领土面积排名世界第几？（C） A、 1 B、 2 C、 3 D、 4 2 【单选题】以下哪个国家不属于...</div>	<div>1137050</div> <div>来自： <a href="#">ling_wang的博客</a></div>
<div>Kotlin实战(三)</div> <div>Kotlin实战(三) 一、元组 1.1、二元元组 data class Pair&amp;amp;lt;out A, out B&amp;amp;gt;{ public val first:...</div>	<div>407</div> <div>来自： <a href="#">zping0808的博客</a></div>
<div>静态分析C语言生成函数调用关系的利器——cflow</div> <div>除了《静态分析C语言生成函数调用关系的利器——calltree》一文中介绍的calltree，我们还可以借助cflow辅助...</div>	<div>4743</div> <div>来自： <a href="#">方亮的专栏</a></div>
<div>FFmpeg详解及常用命令使用</div> <div>FFmpeg简介 FFMPEG堪称自由软件中最完备的一套多媒体支持库，它几乎实现了所有当下常见的数据封装格式、...</div>	<div>20374</div> <div>来自： <a href="#">qq_26464039的博客</a></div>
<div>安装和激活Office 2019</div> <div>有条件请支持正版！相比费尽力气找一个可能不太安全的激活工具，直接买随时随地更新的Office 365确实是最好的...</div>	<div>29144</div> <div>来自： <a href="#">过了即是客</a></div>
<div>java缓冲区</div> <div>1 缓冲区的分类 ByteBuffer CharBuffer ShortBuffer IntBuffer LongBuffer FloatBuffer DoubleBuffer 2 ByteBuffe...</div>	<div>12928</div> <div>来自： <a href="#">weixin_43694144的博客</a></div>
<div>裁员 or 缩招，AI 四起，你的工作还好吗？</div> <div>人工智能时代，AI 已经做了很多“人事”：图片来源：中国新闻网 打扫卫生：roomba 图片来源：wirecutter.com 图...</div>	<div>9101</div> <div>来自： <a href="#">GitChat</a></div>
<div>tcp的java代码</div> <div>服务器端 package com.car.client; import java.io.IOException; import java.io.OutputStream; import java.ne...</div>	<div>12557</div> <div>来自： <a href="#">weixin_43694144的博客</a></div>
<div>webgl第36课-深度冲突-正确处理非常贴近的图形的显示问题</div> <div>需要电子书籍可以Q群：828202939 希望可以和大家一起学习、一起进步！！所有的课程源代码在我上传的资...</div>	<div>1920</div> <div>来自： <a href="#">谷子的博客</a></div>
<div>webgl第24课-几何图形的装配和光栅化</div>	<div>1840</div>



- 各大磁力种子搜索引擎对比

现在磁力种子搜索引擎质量参差不齐，现在就重点整理几个常用的种子搜索站，做个对比分析 1.屌丝搜-最懂屌丝的...

来自： [lizhengnanhua的专栏](#)
- vs2017安装和使用教程（详细）

VS2017如此强大，不仅仅是C语言，Python，ios，Android，Web，Node.js，Azure，Unity，JavaScript等开发都...

来自： [qq\\_36556893的博客](#)
- 数据结构与传统算法总结

快期末了，整理一下笔记，因为是总结嘛，比较杂的知识都没有放，都是和数据结构和传统算法相关的比较重要的...

来自： [hebtu666](#)
- webgl第34课-模型矩阵、视图矩阵、投影矩阵融合

需要电子档书籍可以Q群：828202939 希望可以和大家一起学习、一起进步！！ 上一节课我们学习了 透视投影矩...

来自： [谷子的博客](#)
- webgl第35课-深度缓冲-正确处理图像的位置关系

需要电子档书籍可以Q群：828202939 希望可以和大家一起学习、一起进步！！ 所有的课程源代码在我上传的资...

来自： [谷子的博客](#)

hai836045106

关注

原创33

粉丝30

喜欢6

评论22

等级：

博客4


访问：11万+

积分：1436

排名：4万+



开发c++游戏



最新文章

AndroidStudio 如何关闭 Install Run

Android Studio常用插件

2019人工智能薪资

Python资料免费领

会员任意学

Java薪资多少

怎样才能不被裁员

永久加速器

知网论文查重入口

登录

注册

×

https://blog.csdn.net/hai836045106/article/details/39644215

17/17