

yuxmdef1的专栏

目录视图 摘要视图 RSS 订阅

个人资料



yuxmdef1

访问: 51120次  
积分: 869  
等级: 3  
排名: 千里之外

原创: 38篇 转载: 11篇  
译文: 0篇 评论: 10条

文章搜索

文章分类

- VS2008 (1)
- C (3)
- 计算机基础 (2)
- C++ (7)
- MFC (3)

想听课? 来发话题吧 CSDN APP 博客上线 有奖征文: 云服务器使用初体验 有奖试读—增长黑客, 创业公司必知的“黑科技”

ndk c调用java函数

分类: ndk 2013-12-25 18:13 1926人阅读 评论(0) 收藏 举报

目录(?) [+]

静态函数调用代码

```
[java] C ʹ
01. package com.example.as;
02.
03. import android.os.Bundle;
04. import android.app.Activity;
05. import android.content.res.Resources.Theme;
06. import android.view.Menu;
07. import android.widget.TextView;
08.
09. public class MainActivity extends Activity {
10.
11.     @Override
12.     public void onCreate(Bundle savedInstanceState)
13.     {
14.         try {
15.             Thread.sleep(5000);
16.         } catch (InterruptedException e) {
17.             // TODO Auto-generated catch block
18.             e.printStackTrace();
19.         }
20.         super.onCreate(savedInstanceState);
21.         TextView tv = new TextView(this);
22.         stringFromJNI();
23.         // tv.setText( );
24.         setContentView(tv);
25.     }
26.     public native String stringFromJNI();
```

GDI+ (2)  
孙鑫VC++视频 (8)  
ndk (5)  
二维码 (2)  
android (7)  
swt (8)  
java (1)

文章存档

2015年02月 (1)  
2014年05月 (5)  
2014年03月 (8)  
2014年02月 (3)  
2014年01月 (4)

展开

阅读排行

带默认参数的函数 (4412)  
ZXing简介与核心库编译 (4336)  
CListBox的AddString 排 (3636)  
activity跳转时的生命周期 (3025)  
申请内存的函数 (2863)  
VC中让CListBox带有复选 (2718)  
svn回到历史的某个版本 (2199)  
const变量定义于头文件 (1939)  
ndk c调用java函数 (1922)  
NDK java和c参数传递 (1812)

评论排行

ZXing简介与核心库编译 (3)  
CListBox的AddString 排 (2)  
MipcaActivityCapture代码 (2)  
带默认参数的函数 (2)  
VC中让CListBox带有复选 (1)  
eclipse下32位swt安装 (0)  
swt转exe (0)  
svn回到历史的某个版本 (0)

ndk c调用java函数 - yuxmdef1的专栏 - 博客频道 - CSDN.NET

```
27. public static int add(int x, int y) {  
28.     System.out.println("==Java静态add方法==");  
29.     return x + y;  
30. }  
31. static {  
32.     System.loadLibrary("AS");  
33. }  
34.  
35. }
```

[cpp] C ?

```
01. #include <jni.h>  
02. #include <string.h>  
03. #include <stdio.h>  
04.  
05. int Java_com_example_as_MainActivity_stringFromJNI(JNIEnv* env,  
06.     jobject thiz) {  
07.  
08.     // 获取类  
09.     jclass AnalyzeCidUtil = (*env)->FindClass(env, "com/example/as/MainActivity");  
10.     if (NULL == AnalyzeCidUtil) {  
11.         return -1;  
12.     }  
13.  
14.     // 获取类add静态方法  
15.     jmethodID add = (*env)->GetStaticMethodID(env, AnalyzeCidUtil, "add", "(II)I");  
16.     if (NULL == add) {  
17.         (*env)->DeleteLocalRef(env, AnalyzeCidUtil); // 删除类指引  
18.         return -2;  
19.     }  
20.  
21.     // 调用静态int方法  
22.     int result = (*env)->CallStaticIntMethod(env, AnalyzeCidUtil, add, 3, 5);  
23.  
24.     return 0;  
25. }
```

## 代码解释

java中定义了一个静态函数public static int add(int x, int y)，在C中通过FindClass找到类，GetStaticMethodID找到方法，通过CallStaticIntMethod调用方法  
GetStaticMethodID的第4个参数是什么意思呢？"(II)I"  
这个是方法签名

## 方法签名

在这里有必要单独来讲一讲这个方法签名，为什么要用这个东东呢？我们知道，在Java里方法是可以被重载的。"(II)I"表示的就是这个函数参数为2个int，返回值为int  
其实除了自己对照手写之外，JDK也提供了一个很好用的生成签名的工具javap, cmd进入控制台到你要生成签名的那个类的class文件目录下。如我这里是MainActivity得到结果如下：

[SyncLoadBitmapDemo\(](#) (0)

[ListView优化三原则](#) (0)

推荐文章

- \* Nginx初探
- \* 编译器架构的王者LLVM——  
(6) 多遍翻译的宏翻译系统
- \* 我的第一个Apple Watch小游戏——猜数字 (Swift)
- \* Spark源码阅读：第十节 Standalone运行模式解析
- \* Android基础入门教程——  
8.3.16 Canvas API详解(Part 1)
- \* Android 一个改善的okHttp封装库

最新评论

- [ZXing简介与核心库编译](#)  
[RickyFM](#): 好文要赞, jar包编译出来了, 还没用, 但是额外下载了100M多的maven相关的东西
- [CListBox的AddString 排序问题](#)  
[lin103ning](#): 我也遇到这样问题, 感谢大神的解答
- [MipcaActivityCapture代码阅读---gjhappy](#)  
[gjhappy](#): 调用dialog也会执行surfaceDestroyed 方法
- [MipcaActivityCapture代码阅读---diyangxia](#)  
[diyangxia](#): 这里面对camera的release应该不会有错吧
- [ZXing简介与核心库编译](#)  
[yuxmdef1](#): @sbochao:版本对吗? 我就是这么做的
- [ZXing简介与核心库编译](#)  
[sbochao](#): 这个根本不行啊
- [带默认参数的函数](#)  
[yuxmdef1](#): @ctroll:因为c语言是不支持默认参数的, 当然编译错误了
- [带默认参数的函数](#)  
[ctroll](#): c++可以, c中不可以。  
#include int fun(int a,int b = 1);int ...
- [CListBox的AddString 排序问题](#)  
[wdmcirl](#): 我说难怪了....原来如此  
新手飘过
- [VC中让CListBox带有复选框](#)  
[chinagdlex](#): Owner draw设置为Fixed (LBS\_OWNERDRAWFIXED Has strings...

```
E:\Users\fish\workspace\AS\bin\classes\com\example\as>javap -s MainActivity
Compiled from "MainActivity.java"
public class com.example.as.MainActivity extends android.app.Activity{
    static {};  
    Signature: ()V  
    public com.example.as.MainActivity();  
        Signature: ()V  
    public void onCreate(android.os.Bundle);  
        Signature: (Landroid/os/Bundle;)V  
    public native java.lang.String stringFromJNI(java.lang.String);  
        Signature: (Ljava/lang/String;)Ljava/lang/String;  
    public native java.lang.String unimplementedStringFromJNI();  
        Signature: ()Ljava/lang/String;  
    public static int add(int, int);  
        Signature: (II)I  
}
```

这就是函数和签名的对照表

## 非静态函数调用代码

其实很简单就是多调用一个构造函数构造出一个实例来再去调用非静态函数

```
[java] C 18
01. package com.example.as;
02.
03. import android.os.Bundle;
04. import android.app.Activity;
05. import android.content.res.Resources.Theme;
06. import android.view.Menu;
07. import android.widget.TextView;
08.
09. public class MainActivity extends Activity {
10.
11.     @Override
12.     public void onCreate(Bundle savedInstanceState)
13.     {
14.         try {
15.             Thread.sleep(5000);
16.         } catch (InterruptedException e) {
17.             // TODO Auto-generated catch block
18.             e.printStackTrace();
19.         }
20.         super.onCreate(savedInstanceState);
21.         TextView tv = new TextView(this);
22.         stringFromJNI();
23.         // tv.setText( );
24.         setContentView(tv);
25.     }
26.     public native String stringFromJNI();
27.     public static int add(int x, int y) {
```

```

28.         System.out.println("==Java静态add方法==");
29.         return x + y;
30.     }
31.     static {
32.         System.loadLibrary("AS");
33.     }
34.
35.     /** C回调Java方法（非静态） */
36.     public int sub(int x, int y) {
37.         System.out.println("==Java非静态sub方法==");
38.         return x - y;
39.     }
40.
41. }

```

[cpp] C

```

01. #include <jni.h>
02. #include <string.h>
03. #include <stdio.h>
04. /**
05.  * 实例化类对象
06.  */
07. jobject getInstance(JNIEnv *env, jclass clazz) {
08.     // 获取构造方法
09.     jmethodID constructor = (*env)->GetMethodID(env, clazz, "<init>", "()(V");
10.     if (NULL == constructor) {
11.         return NULL;
12.     }
13.     // 实例化类对象
14.     return (*env)->NewObject(env, clazz, constructor);
15. }
16. int Java_com_example_as_MainActivity_stringFromJNI(JNIEnv* env,
17.     jobject thiz) {
18.
19.     // 获取类
20.     jclass AnalyzeCidUtil = (*env)->FindClass(env, "com/example/as/MainActivity");
21.     if (NULL == AnalyzeCidUtil) {
22.         return -1;
23.     }
24.
25.     // 实例化类对象
26.     jobject mAnalyzeCidUtil = getInstance(env, AnalyzeCidUtil);
27.     if (NULL == mAnalyzeCidUtil) {
28.         (*env)->DeleteLocalRef(env, AnalyzeCidUtil); // 删除类指引
29.         return -2;
30.     }
31.
32.     // 获取对象sub方法
33.     jmethodID sub = (*env)->GetMethodID(env, AnalyzeCidUtil, "sub", "(II)I");
34.     if (NULL == sub) {
35.         (*env)->DeleteLocalRef(env, AnalyzeCidUtil); // 删除类指引
36.         (*env)->DeleteLocalRef(env, mAnalyzeCidUtil); // 删除类对象指引

```

```
37.         return -3;
38.     }
39.
40.     // 调用非静态int方法
41.     int result = (*env)->CallIntMethod(env,mAnalyzeCidUtil, sub, 6, 76);
42.
43.
44.     return 0;
45. }
```

关闭



微信分销系统



微分销系统

王国强博客

微网站

微信分销平台

shell编程学习

加拿大房产

编程学习网

转载。

顶 0 踩 0

主题推荐

java 函数 string c语言 ndk textview native

猜你在找

- Android底层技术: Java层系统服务(Android Service)
- 新版NDK环境搭建免Cygwin超级快
- Apple Watch开发入门
- 编写android程序调用jni本地方法的实现详细例子
- [oeasy]教你玩转 java编程-我的世界mc编程入门
- Android NDK入门
- Android必备的Java基础知识(三)
- Android安装NDK运行环境无需cygwin自动编译生成SO
- JNI: Java与C++的美好结合
- 新版eclipse中NDK环境搭建免Cygwin超级快

查看评论

分享到

- 一键分享
- QQ空间
- 新浪微博
- 百度云收藏
- 微信
- 人人网
- 腾讯微博
- 百度相册
- 更多...

暂无评论

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

核心技术类目

[全部主题](#)   [Hadoop](#)   [AWS](#)   [移动游戏](#)   [Java](#)   [Android](#)   [iOS](#)   [Swift](#)   [智能硬件](#)   [Docker](#)   [OpenStack](#)  
[VPN](#)   [Spark](#)   [ERP](#)   [IE10](#)   [Eclipse](#)   [CRM](#)   [JavaScript](#)   [数据库](#)   [Ubuntu](#)   [NFC](#)   [WAP](#)   [jQuery](#)  
[BI](#)   [HTML5](#)   [Spring](#)   [Apache](#)   [.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](#)   [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 | 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

京 ICP 证 070598 号 | Copyright © 1999-2014, CSDN.NET, All Rights Reserved 