

JNI 技术在桌面搜索工具中的应用

周 强¹, 乐小虬¹, 李 曦²

(1. 中国科学院 国家科学图书馆, 北京 100190;

2. 中国科学技术大学, 安徽 合肥 230026)

摘 要: DocFetcher 是一个用 Java 语言编写的优秀开源桌面搜索工具, 该工具支持的文档格式有 PDF、Word、PowerPoint 等。作者对 DocFetcher 进行了剪裁, 编写了创建索引和检索索引程序。e 划通是用 Visual C++ 开发的, 为了在 e 划通中集成 DocFetcher, 就需要采用 C 语言调用 Java 的技术 (Java Native Interface, 简称 JNI) 来实现。文中首先介绍了 JNI, 接着对该桌面搜索工具 JNI 中 C 部分的程序进行了说明, 分为三部分, 即创建 Java 虚拟机、创建索引、检索索引, 文中列出了主要的程序代码, 并且进行了详细说明。现在, 该桌面搜索工具应用在 e 划通 2.5 完全版中, 功能使用正确, 运行稳定。采用 JNI 作为数据交换的桥梁, 会给软件的实现带来极大的灵活性。C 语言可以在完全体现自身优势的情况下, 整合 Java 语言开发的软件, 更大范围地实现了软件重用。

关键词: Java 虚拟机; Java 本地接口; 桌面搜索

中图分类号: TP39

文献标识码: A

文章编号: 1673-629X(2013)02-0170-03

doi: 10.3969/j.issn.1673-629X.2013.02.043

Applications of JNI Technology in Desktop Search Tool

ZHOU Qiang¹, LE Xiao-qiu¹, LI Xi²

(1. National Science Library, Chinese Academy of Sciences, Beijing 100190, China;

2. University of Science and Technology of China, Anhui 230026, China)

Abstract: DocFetcher is an excellent open source desktop search tool written in the Java language, the tool supports document formats such as PDF, Word and PowerPoint. DocFetcher is trimmed, writing the creation index and retrieval index program. E-pass was developed using Visual C++, in order to integrate DocFetcher with e-pass, need to use C programming language called Java technology (Java Native Interface, referred to as JNI) implementations. It first describes the JNI, then describes the part C of JNI program in the desktop search tool, is divided into three parts, creating Java virtual machine, creating index, retrieving index. It lists the main program code with detailed description. Now, the desktop search tool has applied in the e-pass 2.5 full version, features using correctly, stable operation. Using JNI as a bridge of data exchange, will give great flexibility of software. C language can be entirely in case of its own advantages, integrates software developed by Java language, greater scope to achieve software reuse.

Key words: Java virtual machine; JNI; desktop search

1 JNI 技术分析

JNI 是 Java Native Interface 的简写, 即 Java 本地程序接口, 通过使用 Java 本地接口书写程序, 可以确保代码在不同的平台上方便移植。JNI 使得运行于 Java 虚拟机的代码可以操作 C/C++、汇编等书写的应用程序或库, 同时也可以通过调用相应的接口函数将 Java 虚拟机内嵌到本地应用程序中^[1]。

通常在遇到以下几种情况的时候, 要求程序员利用 JNI 书写本地方法来处理:

1、所开发的应用程序要使用到与平台相关的属性, 而 Java 标准类库不支持对这些属性的处理;

2、已经拥有了用其他编程语言实现的应用程序或库, 希望用 Java 直接调用这些实现;

3、程序的某个模块对运行的时间效率要求很高, 从而希望用较低级的语言 (如汇编) 来实现, 同时希望在 Java 应用程序中使用这个模块;

4、已经拥有了用 Java 实现的应用程序或库, 希望用 C/C++、汇编直接调用这些实现^[2~10]。

2 JNI 技术应用

2.1 桌面搜索工具采用 JNI 技术的背景

e 划通个人研究平台项目计划开发桌面搜索工

收稿日期: 2012-03-27; 修回日期: 2012-07-03

基金项目: 安徽省省级教学研究计划项目 (20100028)

作者简介: 周 强 (1971-), 男, 北京人, 硕士, 主要研究方向为网络信息系统, 信息检索。

具,桌面搜索工具帮助用户搜索定位到本地硬盘中已有文件的内容,分为创建索引、检索索引两部分。为此,对国内外的桌面搜索工具进行了调研,Google、百度等公司推出了各自的桌面搜索工具,开源社区也纷纷推出了桌面搜索工具,DocFetcher是一个用Java语言编写的优秀开源桌面搜索工具,它可以快速地在指定的文件夹搜索特定关键字,该工具支持的文档格式有pdf、word、excel、powerpoint、rtf等。

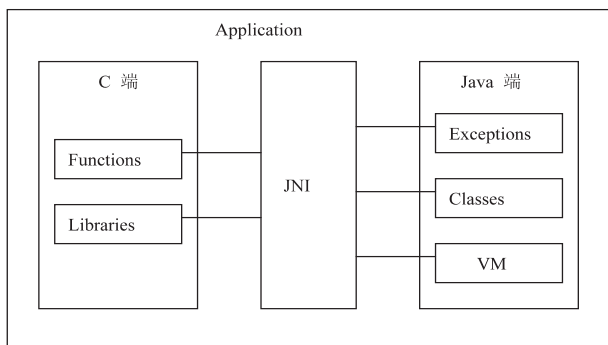


图1 JNI技术分析

e划通是用visual c++开发的,DocFetcher是采用Java语言编写的,启动作者编写的创建索引、检索索引程序有命令行窗口,用户体验不好,用户希望和Windows程序一样启动,这样,就需要采用C语言调用DocFetcher来实现,C程序调用Java程序的技术就是JNI。文中在DocFetcher的基础上进行了裁剪,去掉了图形界面相关的代码^[11~13],采用JNI技术,实现了可供C语言调用的桌面搜索工具。

2.2 C语言调用Java程序的基本步骤

文中,Java集成开发工具是Eclipse,C语言集成开发工具是Visual Studio 2008。

从C语言调用Java代码有4个基本步骤:

- 1、编写Java代码。
- 2、编译Java代码。
- 3、编写C代码。
- 4、运行C应用程序。

下文对桌面搜索工具JNI中C部分的程序进行说明,分为创建Java虚拟机、创建索引、检索索引三部分,为节约篇幅,只写出了主要的程序代码。

2.3 创建Java虚拟机

// JNI 执行环境

JNIEnv * env;

// 指向JVM的指针。主要使用这个指针来创建、初始化和销毁JVM

JavaVM * jvm;

// 具有用于JVM虚拟机的各种选项设置。声明的JavaVMOption options[]数组足够大,能容纳希望使用的所有选项

JavaVMOption options[3];

// 表示可以用来初始化JVM的各种JVM参数

JavaVMInitArgs vm_args;

options[0].optionString = "-Djava.compiler = NONE";

// 虚拟机内存最大设为512M

options[1].optionString = "-Xmx512m";

// 指定Java类的路径,在桌面搜索工具运行环境中,部署了一个精简的Java运行环境,这个运行环境中带有Java系统类。docfetcher.jar是把DocFetcher所有的Java类(包括DocFetcher源代码编译为Java类,所引用的jar解压缩后的Java类)一起压缩的jar包

options[2].optionString = "-Djava.class.path = . ; docfetcher-lib/docfetcher.jar";

vm_args.version = JNI_VERSION_1_4;

vm_args.nOptions = 3;

vm_args.options = options;

// 定义一个函数指针,下面用来指向JVM中的JNI_CreateJavaVM函数

typedef jint (WINAPI * PFuncCreateJavaVM) (JavaVM **, void **, void *);

// 加载JVM.DLL动态库

char * jvmdllPathStr = "jre/bin/client/jvm.dll";

HINSTANCE hInstance = ::LoadLibrary(_T(jvmdllPathStr));

// 取得里面的JNI_CreateJavaVM函数指针

PFuncCreateJavaVM funCreateJavaVM = (PFuncCreateJavaVM) ::GetProcAddress(hInstance, "JNI_CreateJavaVM");

// 调用JNI_CreateJavaVM创建虚拟机

(* funCreateJavaVM) (&jvm, (void **) &env, &vm_args);

2.4 创建索引

// jni中,对应Java类的定义 jclass

jclass cls;

// jni中,对应Java类方法的标识 jmethodID

jmethodID mid;

// 分隔字符串

char * splitChar = ";;";

string splitStr = splitChar;

//定义传递给java的字符串:索引目录名+源文件目录名

本来应该是索引目录名、源文件目录名两个参数,为了简化JNI在C语言和Java语言通信的难度,把这两个参数用特殊字符连接起来,形成了一个字符串

combinParam = indexPath + splitChar + sourcePath;

```

const char * combinParamChar = combinParam.c_
str();
// 查找创建索引的 class 了,因为该类有包路径,
所以要加上包路径
cls = DesktopSearch::env -> FindClass ( " net/
sourceforge/docfetcher/CreateIndexThread" );
//在命令行中,输入 javap - s - p CreateIn-
dexThread,得到 createIndex 的 Signature(签名),作为
GetStaticMethodID ( cls," createIndex "," ( Ljava/lang/
String;)V" )方法的第三个参数。createIndex 是静态方
法,所以用 GetStaticMethodID 调用来取得 createIndex
的 jmethodID 方法标识
mid = DesktopSearch::env -> GetStaticMethodID
( cls, " createIndex ", " ( Ljava/lang/String;)V" );
// 索引目录名+源文件目录名的 windows 字符串
转换为 jstring
jstring arg = NewJString ( DesktopSearch::env,
combinParamChar );
//调用创建索引程序的方法,启动 Java 程序,cre-
ateIndex 是静态方法,所以用 CallStaticVoidMethod 调用
DesktopSearch::env -> CallStaticVoidMethod ( cls,
mid,arg );

```

2.5 检索索引

```

// jni 中,对应 Java 类的定义 jclass
jclass cls;
// jni 中,对应 Java 类方法的标识 jmethodID
jmethodID mid;
//定义动态结构数组,返回检索结果集
struct retMetaData * retValue;
// 查找检索索引的 class 了,因为该类有包,所以
要加上包路径
cls = DesktopSearch::env -> FindClass ( " net/
sourceforge/docfetcher/DoSearcher" );
// 取得 searchIndex 方法的标识
mid = DesktopSearch::env -> GetStaticMethodID
( cls, " searchIndex ", " ( Ljava/lang/String;) [ Ljava/
lang/String;" );
//检索词的 windows 字符串转换为 jstring
jstring arg = NewJString ( DesktopSearch::env,
queryTerm );
//调用检索索引程序的方法,启动 Java 程序,返
回结果集一维对象数组

```

为了简化 JNI 在 C 语言和 Java 语言通信的难度,在 Java 检索程序中,把每个记录的标题、作者、权值、文件路径名、摘要等元数据用特殊字符连接起来,形成一个字符串,这些字符串保存在一维数组中。

```

objectArray msg = ( objectArray ) DesktopSearch::
env->CallStaticObjectMethod( cls,mid,arg );
// 返回结果集对象数组的长度
len = DesktopSearch::env -> GetArrayLength
(msg);
// 分配结果集数组的空间
retValue = ( retMetaData * ) malloc ( sizeof ( struct
retMetaData ) * len+1 );
for(int i=0;i<len;i++) {
// 取得一维对象数组中的元素
object obj = DesktopSearch::env -> GetObjectAr-
rayElement(msg,i);
// 该元素转换为 jstring 对象
jstring str = (jstring)obj;
//jstring 转换为 windows 字符串
char * szStr = JstringToWindows( env,str );
// 从该字符串中提取标题、作者、权值、文件路径
名、摘要等元数据[5,6]
char * haystack=" ";
haystack = szStr;
// 元数据的分隔符
char * needle=" } } ";
int point = 0;
char * buf = strstr( haystack, needle );
while( buf != NULL )
{
point++;
buf[0] = '\0';
if ( point==1 ) {
// 提取标题元数据
retValue[i].title = haystack;
}
...
if ( point==5 ) {
// 提取摘要元数据
retValue[i].sabstract = haystack;
}
haystack = buf + strlen(needle);
buf = strstr( haystack, needle );
}
}
// 返回结果集数组
return retValue;

```

2.6 运行结果

该桌面搜索工具应用在 e 划通 2.5 完全版中,功
(下转第 176 页)

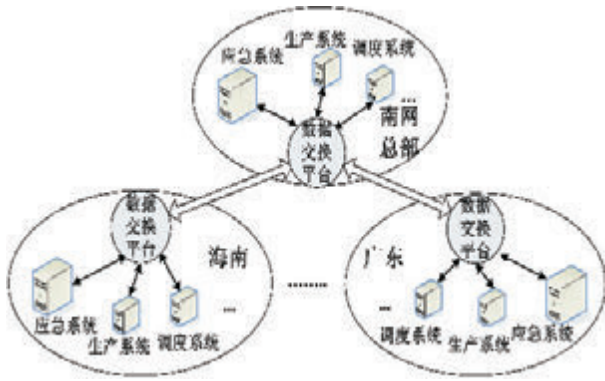


图3 南方电网数据交换分布模式

套安全的数据交换平台,该平台引入整合思想、分层架构思想,以 XML 消息作为交换媒介,以数字信封对 XML 消息进行加密与解密,并以 Web Service 作为 XML 消息的传输手段,实现了业务系统之间快速、安全、可靠的数据交互。该数据交换平台已实际应用于南方电网中。当然,数据交换平台要想能更好地服务企业,就必须能够提供扩展途径适应未来其他的数据格式,通过采用构件化思想,开发新的功能构件,使数据交换平台更加完善^[13]。

参考文献:

- [1] 唐山峰,王淑营.面向电子政务的异构数据交换解决方案[J].计算机技术与发展,2011,21(4):13-16.
- [2] 胡能发,唐为萍.基于 XML 的通用异构数据交换模型[J].计算机工程与设计,2010,31(8):1743-1745.
- [3] 杨甲森,王浩.用于数据交换的 XML 文档和关系数据库转换[J].计算机工程与设计,2006,27(5):857-859.
- [4] 耿飙,宋余庆,梁成全,等.XML 文档到关系数据库映射方法的研究[J].计算机应用研究,2010,27(3):951-954.
- [5] W3C Org. Web Services Description Language (WSDL) Version 2.0[S/OL]. 2007. <http://www.w3.org>.
- [6] 李菲,张新家,袁林.基于 Web Services 的群组数据交换系统的研究与实现[J].计算机技术与发展,2011,21(12):186-190.
- [7] 郝亚东,高敬阳.基于 SOA 的应急数据交换系统设计与实现[J].计算机技术与发展,2010,20(11):205-208.
- [8] 秦贞远,马素霞,齐林海.电能质量数据交换平台的关键问题研究[J].计算机技术与发展,2011,21(4):206-209.
- [9] Siddiqui B. Exploring XML Encryption[EB/OL]. 2002. <http://www.ibm.com/developerworks/xml/library/x-encrypt/>.
- [10] W3C Org. XML Encryption Syntax and Processing[EB/OL]. 2002. <http://www.w3.org/TR/xmlenc-core>.
- [11] 刘艳辉,董碧丹,张峰.数据交换平台的分布式应用研究[J].计算机工程与设计,2009,30(16):3780-3782.
- [12] 孙丕石,曹占峰,王亚玲,等.国家电网公司数据交换平台研发与应用[J].电网技术,2008,32(22):62-67.
- [13] 任开银,孔震,叶敏.通用数据交换系统架构设计与实现[J].电力系统自动化,2009,33(20):62-66.

(上接第 172 页)

能使用正确,运行稳定。

3 结束语

文中首先介绍了 JNI 技术的特点,然后通过桌面搜索工具的实例详细描述了 JNI 技术的使用。采用 JNI 作为数据交换的桥梁,会给软件的实现带来极大的灵活性。C 语言可以在完全体现自身优势的情况下,整合 Java 语言开发的软件,更大范围地实现了软件重用。

参考文献:

- [1] Liron T. Enhance Your Java Application with Java Native Interface[EB/OL]. 1999. <http://www.javaworld.com/javaworld/jw-10-1999/jw-10-jni.html>.
- [2] Janjua W A. Linux JNI Tutorial[EB/OL]. 2003. <http://www.public.asu.edu/~wjanjna/java/jni/>.
- [3] SUN Corp. Java Native Interface[EB/OL]. 2004-01-20. <http://java.sun.com/docs/books/tutorial1/native1>.
- [4] 李亚东,夏雨佳,席裕庚.基于 JNI 的跨平台软件设计[J].计算机工程,2000,26(9):87-88.
- [5] 李东,曹忠升,冯玉才.用 JNI 技术提高 Java 的性能[J].计算机应用研究,2000(10):4-7.
- [6] 沙嘉祥,宁书年,林捷.利用 JNI 实现企业 Java 程序与传统应用程序的集成[J].计算机与现代化,2004(2):23-28.
- [7] 任俊伟,林东岱.JNI 技术实现跨平台开发的研究[J].计算机应用研究,2005(7):180-184.
- [8] 朱朝霞,王杨.Java 本地方法在 Windows 注册表访问中的应用[J].计算机技术与发展,2006,16(11):225-227.
- [9] 杨舰,黄道平,孔钟生.基于 JNI 技术的微机数据提取的研究[J].网络与通信,2007,23(5-3):173-175.
- [10] 王军第,赵凯.JNI 技术在软件开发中的应用研究[J].兰州工业高等专科学校学报,2009,16(5):15-17.
- [11] 周强,李宇,许雁冬.基于 dom4j 转换 XML 为 XHTML 页面的方法[J].计算机技术与发展,2010,20(1):43-45.
- [12] 周强,许雁冬,李宇.应用 Ajax 提高跨库检索系统 SRU 接口的检索速度[J].计算机技术与发展,2010,20(10):203-206.
- [13] 陈勇,刘勇.中医药主题搜索网络机器人的设计与实现[J].计算机技术与发展,2010,20(5):162-166.

JNI技术在桌面搜索工具中的应用

作者：[周强](#)，[乐小虬](#)，[李曦](#)

作者单位：[周强, 乐小虬 \(中国科学院 国家科学图书馆, 北京100190\)](#)，[李曦 \(中国科学技术大学, 安徽合肥230026\)](#)

刊名：[计算机技术与发展](#)

英文刊名：[Computer Technology and Development](#)

年，卷(期)：2013 (2)

本文链接：http://d.g.wanfangdata.com.cn/Periodical_wjtz201302045.aspx