

个人完成部分说明

By 陈锴 (10132510263)

一、实验目的

连接 javaUI 层与底层 C 语言，编写较大项目的 makefile 文件

二、实验内容与实验步骤

1. 研究在 Linux 下 JNI 及 makefile 的使用

2. 查资料，研究网上的 demo

3. 编写代码，整理架构

三、实验环境 Ubuntu 14.04, eclipse

四、完成的主要内容

主要实现了 javaUI 界面与 C 代码的连接还有其他一些功能。

1 Syntab.cc (符号表的实现)

2 Jni_helper.java(连接 UI 和 JNI 的中间层)

3 TestJNI.java(连接 java 代码和 c 代码的中间层)

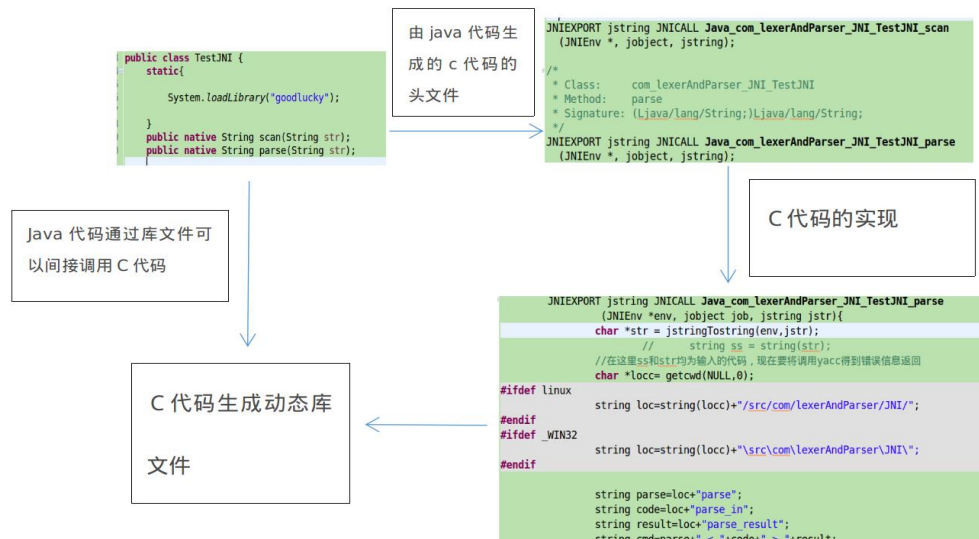
4 com_lexerAndParser_JNI_TestJNI.h(由 java 代码生成的 c 代码的头文件)

5 TestJNI.c(C 代码对扫描和解析的实现部分)

6 Libgoodlucky.so (Linux 下 C 代码的一个动态库，只需包括这个，就可以通过 java 代码调用这个动态库中的 C 代码了)

7 Makefile 文件(Linux 下编译的所需要的文件)

五、 该项目的 JNI (Java Native Interface)的工作原理



六、 makefile 文件说明

```
JAVA=/home/chenkai/Desktop/jdk1.8.0_60
ProDir=/home/chenkai/git/compiler
libgoodlucky.so:makefile TestJNI.o lex.yy.o y.tab.o syntab.o error.o parse
g++ -w -fPIC -rdynamic -shared -o libgoodlucky.so TestJNI.o y.tab.o lex.yy.o syntab.o error.o -ll -ly

TestJNI.o:TestJNI.c com_lexerAndParser_JNI_TestJNI.h
g++ -w -Wall -fPIC -c TestJNI.c -I./ -I $(JAVA)/include -I $(JAVA)/include/linux
```

Makefile 中指明了各个文件之间的编译时的依赖关系，通过简单的 make 命令即可快速进行一个项目的编译工作，因为篇幅原因具体不细说了。

七、 过程中遇到的问题

1 C 代码中的 `yacc` 只适用于命令行的操作，如何将它与 `java` 的 UI 显示联系起来？

解决办法：C 代码调用命令行语句，并将命令行中的结果重定向到文本文件中，再将文本读取到字符串中，并返回给 UI 层。

2 问题 1 解决了，但是命令行中的文本文件需要使用绝对路径，就是说在不同的电脑上需要进行代码的修改及重新编译才可以使用，非常不方便

解决办法：通过 `getcwd()` 函数可以获取所在的文件位置,这样就避免了重复修改和编译的问题

八、心得体会

在 Linux 下好不容易完成了 JNI 的连接工作，我本来是想在 windows 平台下用相同的步骤弄出一个 windows 下可以被调用的 `dll` 文件，然后在这个过程中，我发现 linux 和 windows 的差别实在是太大了，首先文件路径一个是斜杠一个是反斜杠，还有 `flex` 和 `yacc` 在两个平台下有着不同的编译命令。并且我在 linux 下习惯使用命令行操作到了 windows 下却发现 windows 完全不适合使用命令行，连基本的 `g++` 编译命令都要自己去下载 `mingw` 去配置环境，最后我得出一个结论，程序员不适合在 windows 下工作，还是用 linux 吧！