

# 跨平台汇编语言集成系统的设计与实现\*

## Design and Implementation of a Cross-platform Integrated Environment for Assembler Language

蔡启先, 王智文, 陈志成

CAI Qi-xian, WANG Zhi-wen, CHEN Zhi-cheng

(广西工学院计算机工程系, 广西柳州 545006)

(Department of Computer Engineering, Guangxi University of Technology, Liuzhou, Guangxi, 545006, China)

**摘要:**采用 JAVA 面向对象程序设计为主体开发技术,设计实现一个跨 Windows 和 Linux 平台的汇编语言集成系统。该系统以 JAVA 程序中相应的界面视图类、文件检查类、出错信息处理类、编译功能集成类和主类的详细设计来说明系统结构之间的联系,能够实现高级语言程序、底层汇编环境、操作系统环境相互通信,以及 Windows/Linux 双系统兼容,全中文界面交互,源程序编辑、编译、运行及调试集成和出错信息处理,不仅有利于工程技术开发,也有利于初学者掌握汇编语言程序的开发过程。

**关键词:**汇编语言 Windows Linux JAVA

**中图分类号:**TP315 **文献标识码:**A **文章编号:**1002-7378(2008)04-0313-04

**Abstract:** The cross Windows and Linux OS platform integrated environment for Assembler Language is based on objected-oriented JAVA technique. Key technique of the system is identifying OS with KnowWinOrLinux design, identifying and running the assembler files type with CheckFile class, compiling handles for correcting errors with DoErrorMes design, and system running design, procedure corpus (WLasm) design etc. It supports intercommunication within high-level language, assembler language and operating system, compatible Windows/Linux OS, Chinese interface, and source programs editing, compiling, running, debugging and error message processing.

**Key words:** assembler language, Windows, Linux, JAVA

目前,汇编语言程序广泛地用于计算机底层应用的软件开发中。由于基于 DOS 的 Masm 系统存在着界面单调,编辑、编译、连接、调试等功能分立,操作重复性高、人性化操作差等缺点,近年来,人们利用 Windows 平台尝试构建集成化的中文开发环境<sup>[1,2]</sup>,优化了用户操作界面。随着 Linux 平台的兴起,基于 Linux 的汇编语言程序开发同样需要一个实用的集成开发环境。因此,设计一个能兼容 Windows 和 Linux 的跨平台的汇编语言程序集成开发环境很有必要,它不仅有利于工程技术人员高效开发相关程序,也有利于初学者能够尽快掌握汇编

语言程序的开发过程。本文设计一个跨平台汇编语言集成系统,以实现高级语言程序、底层汇编环境、操作系统环境相互通信,以及 Windows/Linux 双系统兼容。

## 1 系统功能及流程

### 1.1 系统功能

跨平台汇编语言集成系统主要具有下述主要功能:(1)操作系统在 Windows 与 Linux 之间变换时,系统会自动识别操作系统并对操作系统进行自适应,包括界面修改、编译集成过程操作的自动调用,调用、生成文件时自动根据不同操作系统区别文件路径。(2)具有 Windows 下的 MASM 系统和 Linux 下的 NASM 系统的全部功能。(3)在不同的操作系统中,都能实现全中文的编辑、编译、连接和运行界

收稿日期:2008-10-12

作者简介:蔡启先(1948-),男,教授,主要从事计算机系统结构、时态数据库的研究。

\* 广西教育厅科研课题(200708LX183)资助。

面。包括：全屏中文界面；编辑界面的智能化处理，包括源程序格式编排处理和字符着色处理，对系统保留字、汇编程序常用格式分别以不同的颜色区分，方便用户输入和改错；编译、连接信息的中英文对照显示和出错位置提示，方便用户程序调试。

### 1.2 系统流程

跨平台汇编语言集成系统处理流程如图1所示，编译集成处理流程如图2所示。

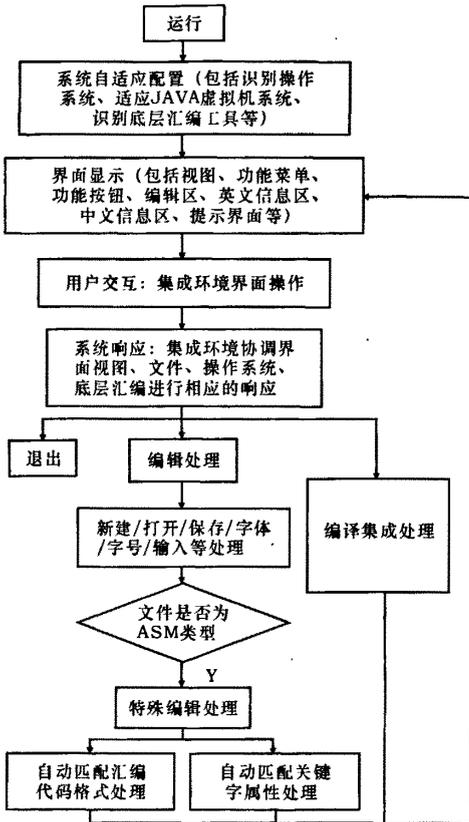


图1 跨平台汇编语言集成系统处理流程

## 2 系统详细设计

跨平台汇编语言集成系统采用 JAVA 面向对象程序设计为主体开发技术，以系统类的详细设计来说明系统结构之间的联系。JAVA 程序中包括 5 个主要的类，分别是：View 界面视图类、CheckFile 文件检查类、DoErrorMessage 出错信息处理类、Asm 编译功能集成类、Wlasm 主类。系统以 Windows 下的 MASM 和 Linux 下的 NASM 为底层汇编编译操作。主要的界面设计、汇编程序风格设计、编译集成设计、编译错误信息获取及处理、数据文件设计、双平台兼容运行设计等，都可以通过 JAVA、MASM、

NASM 三者相互的命令操作和捕获进程间的通信来实现<sup>[3-6]</sup>。

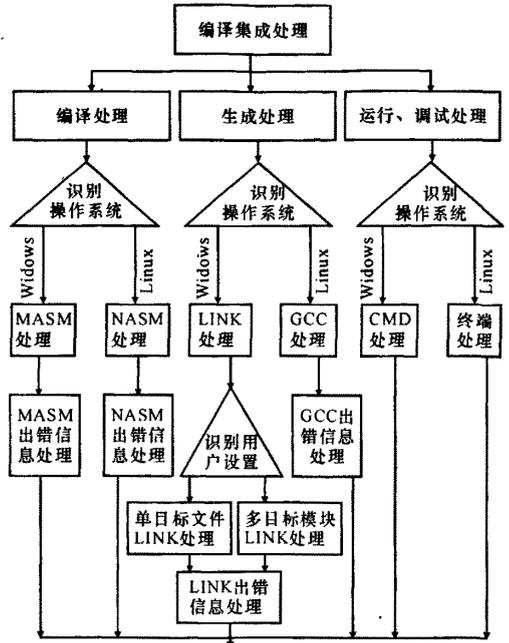


图2 编译集成处理流程

### 2.1 识别操作系统

识别操作系统 KnowWinOrLinux 的设计主要根据 Microsoft Windows OS 家族与 Linux OS 家族在目录设置上的区别。Windows OS 家族目录以盘符字母开始，Linux OS 家族目录以左斜杠“/”开始。获取程序所在目录的绝对路径，判断首字符即可准确识别操作系统。详细代码设计于 KnowWinOrLinux 类构造 KnowWinOrLinux()，以 int 类型字段进行返回，0 为 Windows，1 为 Linux。

### 2.2 识别汇编文件类型并运行有关程序

跨平台汇编语言集成系统在进行编译操作前，首先必须正确的识别将要编译的文件是否为 ASM 类型、是否符合汇编的目录或文件名不超过 8 个 ASCII 字符要求。同样，系统在进行 Windows 平台下的单目标文件生成或多目标模块生成操作前，首先必须正确的识别将要进行生成的文件是否为 OBJ 类型、是否符合汇编的目录或文件名不超过 8 个 ASCII 字符要求。具体做法是，由 View 界面类获得正在编辑的 VFilePath、VFileName，例如：VFilePath = “C:\汇编\Wlasm123\”为汇编非法目录，因为目录包含中文或某级目录超过 8 个 ASCII 字符；VFileName = “例子.txt”为汇编非法文件名，因为文件名包含中文或文件类型不是 ASM。

程序获取 VFileName 最后 3 个字符,分别与需求的类型字符进行比较。VFilePath 去除前面 2 个字符,以(Linux 的)左斜杠“/”或(Windows 的)右斜杠“\”为分隔,获取各级目录单词,将各个单词用正规式“\\w{1,8}”比较,若为非法,则提示(“\\w”表示为可用于标识符的字符元素,紧接的“{1,8}”表示元素可以出现 1 次至 8 次)。详细代码设计于 CheckFile 类方法 CheckAsm()或 CheckObj()。

在识别出汇编文件类型之后,即可由 JAVA 程序产生子程序去调用相关的底层汇编工具 Masm/Nasm、Link/Gcc 等来进行编译/链接。运行程序时,也由 JAVA 程序产生子程序去调用 CMD 命令(Windows 平台): String RunOrder = “start” + FilePath + FileName. substring(0, FileName. length() - 4); 或者在 Linux 平台下由 JAVA 程序产生子程序去调用终端命令: String RunOrder = “./” + FilePath + FileName. substring(0, FileName. length() - 4); 也可以调用 Debug 命令来运行程序: String DebugOrder = “cmd/kdebug” + FilePath + FileName. substring(0, FileName. length() - 4) + “.exe”。

### 2.3 编译出错信息处理

在 Windows 平台下 Masm 编译及 Link 链接而产生信息的文本及其格式,对信息处理所使用的技术主要为编译原理应用、输出重定向、正规式查找、正规式替换、查找数据文件,并产生相应的英文及中文信息。

### 2.4 系统运行设计

使用 JAVA 语言设计的系统,在运行程序时,一般必须以 JAVA 命令模式来运行程序。但是在 Windows 中,根据运行习惯,我们采用 Visual Basic6.0 设计简单程序 Wlasm.exe 来实现鼠标双击引导运行;在 Linux 中,根据运行习惯,将程序设计为在 Linux 终端使用 JAVA 命令运行程序。

### 2.5 程序主体设计

Wlasm extends View,用面向对象思想,承继、重载或增加字段和方法,增加编译集成的界面交互的菜单和按钮。增加编译集成的设置项,用于设置是否产生中间文件、是否进行多目标模块生成。

## 3 系统实现

跨平台汇编语言集成系统在编辑、编译和连接汇编语言程序中的应用情况如图 3、图 4、图 5 所示。

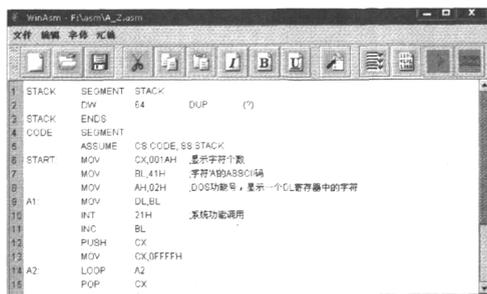


图 3 ASM 类型文件的打开界面



图 4 Windows 平台下编译过程及出错信息处理输出

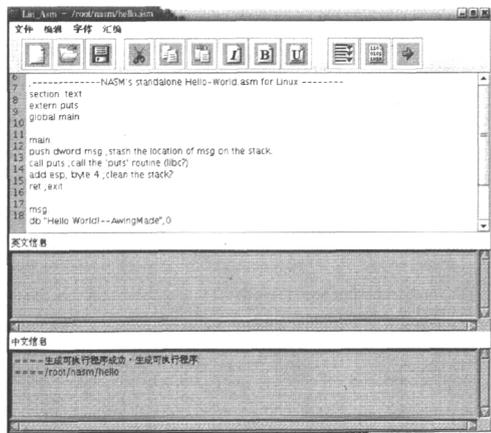


图 5 Linux 平台下的生成可执行程序及信息输出

图 3 显示 ASM 类型文件的打开界面,系统自动对源程序进行格式编排、语句编号及关键字着色处理,界面清晰,输入与编程方便。图 4 显示在 Windows 平台下的编译过程,若程序有语法错误,出错信息输出采用中英文对照显示,方便编程者观察和理解出错信息,Linux 系统中同样如此。图 5 显

示在 Linux 平台下通过对目标程序连接后生成了可执行程序及信息输出,同样采用中英文对照显示,使编程者易于理解。Windows 平台下也具有类似的界面。无论在 Windows 或 Linux 环境下,系统都能新建、打开、编辑、保存各种不同的文件,包括 ASM、TXT、TEXT 等类型或无类型的文件。

#### 4 结束语

随着 Linux 应用的越加广泛,采用 Windows、Linux 双系统兼容设计的软件将会越来越多,本文设计实现的跨平台汇编语言集成系统,能够兼容 Windows 和 Linux 系统平台,该系统不仅有利于工程技术开发,也有利于初学者掌握汇编语言程序的开发过程。目前该系统已经在汇编语言教学中应用,它以其方便实用的特点得到了普遍欢迎。

#### 参考文献:

- [1] 刘跃华,梁英. 智能化的汇编语言集成开发环境[J]. 计算机技术与自动化,2007,26(3):114-117.
- [2] 沈洁萍,章红,陈勇. 汇编语言集成编译环境的开发[J]. 微计算机信息,2004,20(6):120-121.
- [3] 蔡启先,王智文,黄晓璐. 汇编语言程序设计实验指导[M]. 北京:清华大学出版社,2008.
- [4] 陈火旺,刘春,谭庆平,等. 程序设计语言编译原理[M]. 第3版. 北京:国防工业出版社,2000.
- [5] 耿祥义,张跃平. Java2 实用教程[M]. 北京:清华大学出版社,2006.
- [6] 马季兰,彭新光. Linux 操作系统[M]. 北京:电子工业出版社,2002.

(责任编辑:韦廷宗)

(上接第 312 页)

#### 3 实验分析

构造一个简单的样本集,样本由 60 个网页组成。使用 30 个与 NBA 有关信息相关的网页构成前 30 个样本,使用另外 30 个与 NBA 不相关的网页构成后 30 个样本。类别分成两类,一类是 NBA 类,一类是其他类。利用 KNN 分类程序对待分类文本进行分类,观察分类结果。

在 60 个待分类文本中,NBA 类的准确率是 90%。其他类的准确率是 76.67%。分类时间大约 3s。在实验环境下,样本中的 NBA 类可以看作是“带有不安全信息”的网页类,其他类则看作是安全的网页类。能够正确分析出某一待分类网页是否为 NBA 类,则说明分类器的工作是有效的。

#### 4 结束语

本文对网页的安全性进行挖掘,设计实现一个基于 Web 挖掘的自动分类器,并构造一个实验环境来检测分类器的性能。实验结果表明,所设计的自动

分类器可以得到比较好的分类结果。但是由于 KNN 属于惰性分类方法,所以耗费的资源较大,速度相对较慢。我们下一步的工作将继续研究 Web 挖掘中关于网页安全性的判断和检测方法,进一步提高自动分类器的分类准确率,在准确率和速度之间寻求平衡点。

#### 参考文献:

- [1] Han Jiawei, Micheline Kamber. 数据挖掘:概念与技术[M]. 第2版. 北京:机械工业出版社,2007.
- [2] 冯迪,李晋宏,曹原. 基于网页的数据挖掘研究:2007 通信理论与技术新发展——第十二届全国青年通信学术会议论文集:上册[C]. 北京:电子工业出版社,2007.
- [3] 张华平,刘群. 基于 N-最短路径方法的中文词语粗分模型[J]. 中文信息学报,2001(5):1-7.
- [4] 梁循. 数据挖掘算法与应用[M]. 北京:北京大学出版社,2006.

(责任编辑:韦廷宗)