**中国矿业大学计算机学院**

**2019级本科生课程设计报告**

课程名称 系统软件开发实践

报告时间 2022年2月25日

学生姓名 胡钧耀

学 号 06192081

专 业 计算机科学与技术

任课教师 张博

**成绩考核**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 编号 | 课程教学目标 | | | 占比 | | 得分 |
| 1 | **目标1：**针对编译器中词法分析器软件要求，能够分析系统需求，并采用FLEX脚本语言描述单词结构。 | | | 15% | |  |
| 2 | **目标2：**针对编译器中语法分析器软件要求，能够分析系统需求，并采用Bison脚本语言描述语法结构。 | | | 15% | |  |
| 3 | **目标3：**针对计算器需求描述，采用Flex/Bison设计实现高级解释器，进行系统设计，形成结构化设计方案。 | | | 30% | |  |
| 4 | **目标4：**针对编译器软件前端与后端的需求描述，采用软件工程进行系统分析、设计和实现，形成工程方案。 | | | 30% | |  |
| 5 | **目标5：**培养独立解决问题的能力,理解并遵守计算机职业道德和规范，具有良好的法律意识、社会公德和社会责任感。 | | | 10% | |  |
| 总成绩 | | | | | |  |
| 指导教师 | |  | 评阅日期 | |  | |

**目 录**

[实验（二） 利用 Flex 构造编译器 1](#_Toc96678600)

[2.1 实验目的 1](#_Toc96678601)

[2.2 C语言子集C1的定义 1](#_Toc96678602)

[2.3 *lex2-1.l*编写 1](#_Toc96678603)

[2.3.1 题目要求 1](#_Toc96678604)

[2.3.2 分析步骤 1](#_Toc96678605)

[2.3.3 文件源代码 1](#_Toc96678606)

[2.3.4 双系统实验结果 1](#_Toc96678607)

[2.4 *lex2-2.l*编写 1](#_Toc96678608)

[2.4.1 题目要求 1](#_Toc96678609)

[2.4.2 分析步骤 2](#_Toc96678610)

[2.4.3 文件源代码 2](#_Toc96678611)

[2.4.4 双系统实验结果 2](#_Toc96678612)

[2.5 注意事项 2](#_Toc96678613)

[2.5.1 命令行的选择 2](#_Toc96678614)

[2.6 课后练习 3](#_Toc96678615)

[2.6.1 题目 3](#_Toc96678616)

[2.6.2 解答 3](#_Toc96678617)

[2.7 实验感悟 3](#_Toc96678618)

1. 利用 Flex 构造编译器
   1. 实验目的

阅读*Flex/Bison.pdf*第一章，第二章，掌握Flex基础知识。利用Flex实现用于C语言子集C1的词法分析器。

* 1. C语言子集C1的定义

该子集主要由关键字、专用符号、标识符、整型常数、空格、注释构成。下面一一介绍各个部分。

关键字：所有的关键字都是保留字，并且必须是小写。

else if switch for int float return void while

专用符号。

+ - \* / < <= > >= == != = ; , ( ) [ ] { }

标识符（ID）和整型常数（NUM）通过下列正则表达式定义（注：小写和大写字母是有区别的）。

L = [a-zA-Z\_]

D = [0-9]

ID = {L} ({L} | {D})\*

NUM = [1-9] {D}\*

空格由空白、换行符和制表符组成。

[space] \n \t

注释有单行注释和多行注释。

// /\*……\*/

* 1. *lex2-1.l*编写
     1. 题目要求

编写Flex代码*lex2-1.l*，实现对上述C1语言的词法分析。要求：输出所有的关键字KEYWORD、专用符号SYMBOL、标识符ID、整型常数NUM。

* + 1. 分析步骤
    2. 文件源代码
    3. 双系统实验结果
  1. *lex2-2.l*编写
     1. 题目要求

在实现以上基本功能的基础上，参考*ANSI C grammar (Lex).pdf*，实现以下功能，并另存为*lex2-2.l*文件：

输出上述标记所在的行号row、列号col；

忽略注释及其内容，如，注释中的数字/\*123\*/ , //123；

增加科学记数法；

十六进制、八进制常数。

* + 1. 分析步骤
    2. 文件源代码
    3. 双系统实验结果

表 1 手工验证Windows系统lex1输出结果

| 匹配 | 字符 | 单词 | 行 | 匹配 | 字符 | 单词 | 行 |
| --- | --- | --- | --- | --- | --- | --- | --- |
| #include | 8 | 1 |  | "<<endl | 7 | 1 |  |
| [space] | 1 |  |  | \n | 1 |  | 1 |
| iostream | 8 | 1 |  | cout | 4 | 1 |  |
| \n | 1 |  | 1 | [space] | 1 |  |  |
| using | 5 | 1 |  | "Welcome | 8 | 1 |  |
| [space] | 1 |  |  | [space] | 1 |  |  |
| namespace | 9 | 1 |  | to | 2 | 1 |  |
| [space] | 1 |  |  | [space] | 1 |  |  |
| std | 3 | 1 |  | c++! | 4 | 1 |  |
| \n | 1 |  | 1 | [space] | 1 |  |  |
| int | 3 | 1 |  | " | 1 | 1 |  |
| [space] | 1 |  |  | [space] | 1 |  |  |
| main | 4 | 1 |  | endl | 4 | 1 |  |
| \n | 1 |  | 1 | \n | 1 |  | 1 |
| cout | 4 | 1 |  | return | 6 | 1 |  |
| [space] | 1 |  |  | \n | 1 |  | 1 |
| "Hello! | 7 | 1 |  | 合计 | 104 | 17 | 6 |
| [space] | 1 |  |  |  |  |  |  |

基于Linux进行的手工分析可以在表 1基础上分析，共出现6个\n，应该会读取到六个\r\n，对于Windows的结果就应该给字符数加6，其他不变，即字符数110，单词数17，行数16，该结果与图 7程序运行结果一致，验证正确。

* 1. 注意事项
     1. 命令行的选择

Developer PowerShell for VS 2022 不支持输入含“<”的命令。需要使用Developer Command Prompt for VS 2022。

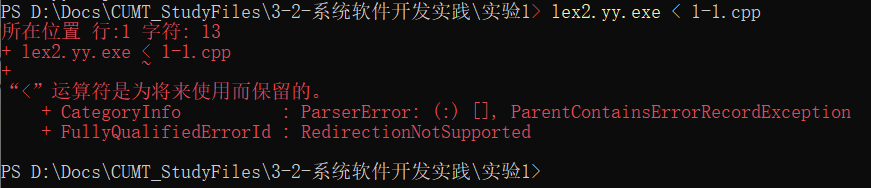


图 9 选择合适的命令行工具

* 1. 课后练习
     1. 题目

给定如下的匹配规则，对于表达式i+=1，是+、=分别输出还是一起输出？

“+” {printf(“+”);}

“=” {printf(“=”);}

“+=” {printf(“+=”);}

* + 1. 解答

哈哈哈

* 1. 实验感悟

本次实验，