**需求分析文档**

# 一、引言

## 1.1目的

该文档是本小组为开发mini-C语言解释器而编写的需求分析报告，为解释器的设计及开发工作提供可靠的依据。

## 1.2项目背景

1. 项目名称：mini-C 解释器；
2. 本项目的任务提出者：陈硕鹰老师；
3. 本项目的完成者：端茶倒水小组；
4. 产品用户：mini-C程序编写者；
5. 编程语言：C++；
6. 开发环境：VS；
7. 遵循的规范：见《mini-C语言语法说明》、《小组命名规范和代码规范》；
8. 测试环境：Windows10；
9. 软件交付日期：2017/9/14前。

## 1.3词语定义

1. 解释器：即本小组编写的mini-C语言解释器程序；

## 1.4文档综述

项目需求分析报告包含以下多个部分：引言、具体需求、需求建模、需求分级以及附录。

1. 引言：明确项目需求分析报告的书写目的，确定软件范围，提供书写参考资料。
2. 具体需求：详细描述项目的具体需求，包括：功能需求、接口需求、性能需求、软件质量要求。
3. 需求建模：提供一种详细说明系统的结构或行为的方法，帮助我们按照实际情况或按我们需求的样式对系统进行可视化。
4. 需求分级：按照需求的重要性进行整理和分级，指示不同需求的开发紧急程度。
5. 附录：撰写本文档的其他参考资料。

# 二、具体需求

## 2.1功能性需求：

### 2.1.1输入

#### 2.1.1.1输入文本

用户将自己需要解释的程序以文件或命令行形式导入至解释器中。

#### 2.1.1.2词法分析

从左到右逐个字符的读入用户提交的源程序，即对构成源程序的字符流进行扫描然后根据词规则识别单词（或单词符号或符号）。

#### 2.1.1.3 产生保留字集

确定mini-C语言中的关键字，建立词法集。

根据现有的词法集，针对读入文本产生保留字。

将所有的保留字归并到一个集合中。

### 2.1.2识别

#### 2.1.2.1 确定代码块并解析代码块

**构造语法树**：对代码块中的表达式进行构造语法树。

**遍历形成后缀表达式**：遍历语法树，将表达式构变成后缀表达式。

**解析表达式：**计算后缀表达式。

**形成代码块：**将表达式的值构成代码块。

**控制语句解析：**将代码块中的控制语句解析形成语法树。

#### 2.1.2.2 函数分析

**确定局部变量生命周期：**确定函数中的局部变量，生命周期，方便申请内存。

**确定返回值类型：**确定函数的返回值，同时记录函数的返回值类型 。

**确定形参：**确定函数的输入形参，记录函数的形参个数以及类型。

**确定代码块并解析代码：**将函数内的代码块执行2.1.2.1的操作。

### 2.1.3执行中间代码

在分析语法，解析语法的同时一边执行运算，同时生成解释信息。

### 2.1.4输出解释信息

根据每部分不同的功能确定出错信息，建立异常库。在嵌套的形式中，每一层向上一层抛出异常信息，最后最外层输出。

#### 2.1.4.1解释成功信息

无致命错误，整个程序正常解释成功。

#### 2.1.4.2输入错误信息

输入文件不存在或格式错误。

#### 2.1.4.3词法错误信息

词法分析错误，关键字拼写错误。

#### 2.1.4.4内存分配错误信息

内存分配出错，无法分配或尝试给常量分配

#### 2.1.4.5表达式错误信息

表达式计算出错，表达式格式出错。

#### 2.1.4.6语法错误信息

关键字不匹配或无语句结束符“；”。

## 2.2非功能性需求：

### 2.2.1 可靠性

1）可用性：在后期维护期间，用户也可同时使用。

2）健壮性：如果用户在没有完全保存自己的数据时或者由于其他因素，用户和系统的连接中断，那么用户可以在系统中恢复自己已编程序代码的数据。

### 2.2.2 易用性

1）系统面对的是具备简单编程能力的人员，因此，在事物描述上可用术语描述即用术语描述，避免产生二义性，让用户产生歧义。

2）系统在设计时，所用代码风格较为标准，在后期维护上更为方便。

### 2.2.3安全

1）所有涉及功能信息和代码信息的数据都要进行加密操作。系统安全级别应该是强制保护级。

2）用户可以设置自己代码数据的可见性

### 2.2.4可扩展

在现有功能的基础上，可以进行其他功能的扩展。例如可以增加系统可以识别的语法，增加可识别多态性的功能等。

### 2.2.5性能

本程序在要求的100行Mini-C代码，不超过5层循环嵌套的情况下，至少做到一般的Mini-C代码解释时间不超过2s。

## 2.3可选需求：

### 2.3.1 UI（用户界面）

#### 实现方式：

HTML + C++ Socket

#### 功能需求：

1. 上传代码文件或文本

2. 展示解释结果

#### 细分功能需求：

1. 完成各个终端设备的文件选择

2. 支持各种格式的内容输入

3. 展示上传文件中的内容

4. 展示服务器回传的程序解释结果，并对应到行

# 三、需求建模



数据流图-0层图



数据流图-1层图



数据流图-读入-2层图



数据流图-识别-2层图



数据流图-识别-3层图



数据流图-输出解释信息-2层图

# 四、需求分级

## 一级需求：

* 输入mini-c程序文本
* 分析文本识别单词
* 对mini-c程序进行词法分析
* 对形成的代码块进行语法分析
* 进行函数分析
* 分析的同时进行运算
* 输出mini-c程序结果

## 二级需求：

* UI界面设计
* Mini-c程序的调试功能
* 输出错误信息或者解释信息

## 三级需求：

* 易用性（见非功能需求-易用性需求）
* 可靠性（见非功能需求-可靠性需求）
* 用户在编写代码时，自动对代码的相关数据进行加密

## 四级需求：

* 长代码在短时间内生成结果
* 多次循环能够正常运行

## 五级需求：

* 良好的用户交互体验
* 自动代码补全

# 五、附录

## 5.1参考资料

[1] mini-C语言语法说明

[2]CSDN博客 <http://www.cnblogs.com/linxr/>

[3]两周自制脚本语言，千叶滋[日]