中文编译器规划书

一、目标

实现一门用中文来编程的解释型语言，支持中英文，利用ANSI格式来存储的文本文件来编程，形成中文解释型语言。

二、支持关键字

关键字采用”<”,”>”来作为界定符号，这两个符号支持关键字和复杂标识符。具体支持的关键字如下表所列：

|  |  |  |
| --- | --- | --- |
| 关键字符号 | 用途 | 存储长度 |
| <如果> | 条件语句 | 6 |
| <直到> | 循环语句 | 6 |
| <全局量> | 用于函数定义时内部变量的全局申明 | 8 |
| <返回> | 用于函数定义时返回用 | 6 |
| <真> | 用于定义逻辑常量真 | 4 |
| <假> | 用于定义逻辑常量假 | 4 |

三、支持运算符号和标识符号

|  |  |  |  |
| --- | --- | --- | --- |
| 符号 | 名称 | 功能 | 存储长度 |
| + | 加号 | 加法 | 1 |
| - | 减号 | 减法和负运算 | 1 |
| \* | 星号 | 乘法 | 1 |
| / | 除号 | 除法取商 | 1 |
| % | 百分号 | 除法取余 | 1 |
| && | 逻辑且符号 | 逻辑且运算 | 2 |
| || | 逻辑或符号 | 逻辑或运算 | 2 |
| = | 等号 | 赋值运算 | 1 |
| > | 大于号 | 逻辑判断运算 | 1 |
| < | 小于号 | 逻辑判断运算 | 1 |
| <= | 小于等于 | 逻辑判断运算 | 2 |
| >= | 大于等于 | 逻辑判断运算 | 2 |
| == | 是等于 | 逻辑判断运算 | 2 |
| != | 不等于 | 逻辑判断运算 | 2 |
| ： | 冒号 | 用于函数定义 | 1 |
| { | 左大括号 | 语句集合包含用 | 1 |
| } | 右大括号 | 语句集合包含用 | 1 |
| ( | 左小括号 | 用于表达式 | 1 |
| ) | 右小括号 | 用于表达式 | 1 |
| ; | 语句分号 | 用于一条语句识别 | 1 |

四、标识符

标识符包括两种模式

1、字符模式：以字符开头，包括英文字符，中文字符两种，英文字符A-Z，a-z，中文字符高字节0xA1-0xF7，低字节0xA1-0xFE

2、符号模式：以”<”开头，以“>”结尾，中间可以包含任意字符，但空格，等号，大小括号,以及大于小于号除外，该模式一定要求出现”>”号，不然解释系统要求报错，并提示用户。

标识符系函数名称或变量名称。变量不在同一函数的，为不同的变量。

函数为默认全局，内置函数使用符号模式的标识符。

五、应用举例

代码一:（打印函数即为解释器内置函数）

<打印>(“你好，中文编程！”);

代码二：

打印数字：(数字){

<打印>(“输出的数字为”+数字);

<返回>0;

}

返回值 = 打印数字(5);

打印数字(返回值);

代码一，屏幕输出“你好，中文编程！”

代码二、屏幕输出“50”

六、常量支持

1、整数常量（仅支持十进制：即0-9开头。尾0-9）

2、实数常量（仅支持十进制：即0-9开头。尾0-9，外加一个小数点）

3、逻辑真假常量（关键字真和假）

4、字符串常量支持（双引号开头，双引号结尾）；

七、解释器设计（C语言版本）

结构设计

1、单词：

struct Token{

char \* tokenaddress;

int tokenlen;

};

2、常量(执行时动态分配内存空间，识别出后先动态分配)

struct Constant{

ConstantType type;//常量类型，包括四种

int size;//量的空间

struct Constant \* nextConstant;//下一个常量的地址

int count;//引用计数，计数为零的从表中剔除

char[] data;

};

3、变量

struct Variable

{

Token \* name; // 指向名称的单词地址

struct Constant \* value;//变量值，在执行过程中变化， 在执行前为NULL

};

4、表达式

一元表达式｛

常量表达式：一个常量指针

变量表达式：一个变量指针

函数调用表示式：一个函数名（Token）指针和参数列表指针（表达式列表）

复合表达式: 一个表达式指针

｝

取反表达式：一个一元表达式指针

乘除法表达式：一个乘除法表达式指针和一个取反表达式指针

加减法表达式: 一个加法表达式指针和一个乘除法表达式指针

比较表达式：两个加减法表达式指针

逻辑或且运算表达式：一个逻辑或且运算表达式指针和一个比较表达式指针

赋值表达式指针：一个变量表达式指针和一个逻辑或且运算表达式指针。

5、语句

表达式语句：赋值表达式指针或函数调用表达式指针

条件语句：一个逻辑或且运算表达式指针和语句列表指针

循环语句：一个逻辑或且运算表达式指针和语句列表指针

返回语句：一个逻辑或且运算表达式指针

全局申明语句：一个Token地址

6、变量列表（用于全局变量记录）

一个变量地址，一个下一个列表项的地址

7、表达式列表

一个表达式指针，一个下一个表达式列表项地址

8、语句列表

一个语句地址，一个下一个语句项地址

9、自建函数 一个Token地址（函数名称）、一个形参列表地址（变量列表）、一个语句列表指针

内置函数一个Token地址（函数名称）、一个形参列表地址（变量列表）

函数（函数类型、函数对象）

10、函数列表

一个函数地址，一个下一个函数项地址

11、解释器

函数列表

语句列表

变量列表

1、词法分析器（函数）

1）、State ScanToken(IN char \*filedata, IN int filedatalen,IN Token \* begintoken,OUT Token \* nexttoken);

解释：filedata：文件数据

filedatalen：文件数据长度

begintoken：当前词

nexttoken：用于输出，下一个词

2）、State ReadSrcFile(char \* filepath);

解释：filepath：文件路径

struct Token{

char \* tokenaddress;

int tokenlen;

};

3）、State CompareToken(Token\* src,Token\* dsc);

src: 代表目标

dsc：代表比较对象

解释器函数：

1、编译

2、执行

3、初始化

“函数”函数

1、创建

2、执行

“函数表”函数：

1、增加

2、查询

语句列表函数

1、增加

2、执行

“变量表”函数

1、增加

2、查询

表达式函数

1、创建

2、执行

常量操作函数：

1、创建增加

2、清理常量表

变量操作函数：

1. 创建函数

ANSI 字符范围：’A’-‘Z’ ‘a’-‘z’

中文范围： 高区字节0xB0-0xF7(16区-87区) 低位字节0xA1-0xFE(94个字)

词法分析：

词汇支持：