|  |
| --- |
| 华东师范大学 计算机科学与技术系 |
| Small语言编译器 |
| 测试说明书 |

|  |
| --- |
| 毛杰文 10102130253  2013-10-23 |

# 概述

## 测试描述

根据已开发的small语言编译器，使用它编译用small语言写的两个程序，验证编译器设计的正确性。

## 测试环境

硬件环境：普通PC

软件环境：操作系统：Windows 7。要求安装.NET Framework 3.5以上版本。

# 测试用例描述

## 2.1 测试1：输入两个正整数，求它们的最大公约数。

### 2.1.1 测试目标

1、 能去掉程序内的注释。

2、 能够编译出中间代码。

3、 解释执行中间代码时，能得到正确的步骤和结果。

### 2.1.2 测试用例源程序

/\* Program 1: Find 2 number's GCD \*/

a := 0;

b := 0;

read a;

read b;

if a<b then // if a is less than b, change a and b

begin

t := b;

b := a;

a := t;

end

repeat

begin

r := a%b;

a := b;

b := r;

end

until r == 0;

write a;

### 2.1.3 测试步骤

1、 执行smallcc test1.txt test1out.txt。看一下是否能输出中间代码；是否能去掉原程序中的注释；是否能找出语法错误。

2、 执行smallint test1out.txt。看一下是否能根据中间代码得出正确的结果。

3、 执行smallint test1out.txt –s –l 看一下能否输出数据栈和当前指令。

### 2.1.4 测试结果

生成的中间代码如下：

push 0

pop a

push 0

pop b

push a

in a

push b

in b

push a

push b

compLT

jz L000

push b

pop t

push a

pop b

push t

pop a

L000:

L001:

push a

push b

mod

pop r

push b

pop a

push r

pop b

push r

push 0

compEQ

jz L001

L002:

push a

out

输出结果如下：（下划线为输入内容）

a? 10

b? 4

2

以及

a? 150

b? 90

30

能够编译并运行出正确的结果，说明该编译器运行状况良好。

## 2.2 测试2：求1-100以内的素数

### 2.2.1 测试目标

测试新增加的运算符，例如++,--以及数组的定义和使用。

### 2.2.2 测试用例源程序

/\* Program 2: Find all primes between 1 and 100 \*/

array prime[101];

/\* Initialize the array \*/

i := 2;

while(i<101) do

begin

prime[i] := 1;

i++;

end

/\* Set all prime 1 and non-prime 0 \*/

i := 2;

while i<101 do

begin

if prime[i] == 1 then

begin

j := 2;

while j <= 100/i do

begin

prime[j\*i] := 0;

j++;

end

end

i++;

end

/\* output \*/

i:=1;

while(i<101)do

begin

if prime[i] == 1 then write i;

i++;

end

### 2.2.3 测试步骤

同2.1.3，但由于100个数组的堆栈比较大，因此不再进行单步运行测试。

### 2.2.4 测试结果

生成的中间代码如下：

new prime

alloc 101

push 2

pop i

L000:

push i

push 101

compLT

jz L001

push prime

push i

push 1

stm prime

push i

inc

jmp L000

L001:

push 2

pop i

L002:

push i

push 101

compLT

jz L003

push prime

push i

ldm prime

push 1

compEQ

jz L004

push 2

pop j

L005:

push j

push 100

push i

div

compLE

jz L006

push prime

push j

push i

mul

push 0

stm prime

push j

inc

jmp L005

L006:

L004:

push i

inc

jmp L002

L003:

push 1

pop i

L007:

push i

push 101

compLT

jz L008

push prime

push i

ldm prime

push 1

compEQ

jz L009

push i

out

L009:

push i

inc

jmp L007

L008:

运行结果如下：

2

3

5

7

11

13

17

19

23

29

31

37

41

43

47

53

59

61

67

71

73

79

83

89

97

可见能输出正确的结果。说明数组的定义和使用没有问题。