**综合测试报告**

目录

[1.测试用例 2](#_Toc5807)

[2.测试结果分析： 3](#_Toc5052)

[2.1斐波那契数列： 3](#_Toc25755)

[2.1.1测试代码： 3](#_Toc25497)

[2.2.2汇编代码 4](#_Toc29884)

[2.2.3运行结果： 7](#_Toc11824)

[2.2冒泡排序 8](#_Toc13743)

[2.2.1测试代码 8](#_Toc21581)

[2.2.2 汇编代码 9](#_Toc10001)

[2.2.3运行结果 16](#_Toc30999)

[3. sysy测试用例 16](#_Toc24965)

[4. 总结 17](#_Toc5232)

测试用例分为两部分，一个部分是自己编写的测试用例，用于检查实现的功能，并对错误的地方进行debug修改，在各个模块的测试中，对已实现的功能达到了覆盖。另一部分是2020年编译器设计决撒中sysy的测试用例，因为sysy规定的文法有地方没有实现，因此没有通过比赛的测试用例。

选用了自己设计的两个综合测试程序，递归斐波那契数列和冒泡排序，以及sysyruntime中的测试用例进行综合测试。

其他的模块测试包括

|  |  |
| --- | --- |
| 测试模块 | 测试内容 |
| 词法分析测试 | token序列 |
| 语法分析，符号表，中间代码生成测试 | 错误信息，符号表生产，中间代码生成 |
| 中间代码优化测试 | 优化后的中间代码 |

模块的测试报告详见该目录下的其他文档。

# 1.测试用例

|  |  |  |
| --- | --- | --- |
| 测试代码 | 是否通过 | 错误原因分析 |
| bubble.cpp | 通过 |  |
| fib.cpp | 通过 |  |
| conv0.sy | 未通过 | 数组作为函数参数没有实现  条件表达式短路没有实现 |
| fft0.sy | 未通过 | 数组作为函数参数没有实现  条件表达式短路没有实现  取模运算 |
| median0.sy | 未通过 | 数组作为函数参数没有实现  条件表达式短路没有实现 |
| shuffle0.sy | 未通过 | 数组维度可以是常量表达式 |
| transpose0.sy | 未通过 | 数组作为函数参数没有实现  条件表达式短路没有实现  初始化除法运算错误 |

# 2.测试结果分析：

## 2.1斐波那契数列：

### 2.1.1测试代码：

|  |
| --- |
| int ret;  int Fibon1(int n)  {      if (n == 1)      {          return 1;      }      if(n==2)      {          return 1;      }      else      {          return Fibon1(n - 1) + Fibon1(n - 2);      }  }  int main()  {      int n = 6;      ret = Fibon1(7);      putint(ret);      return 0;  } |

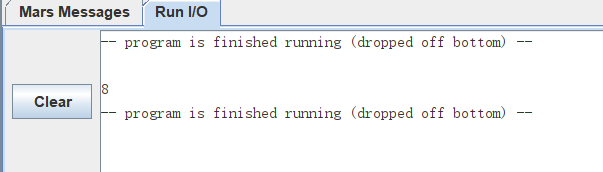
### 2.2.2汇编代码

|  |
| --- |
| .data  ret: .word -858993460  .text  j main  Fibon1:  sw $ra,0($sp)  sw $fp,-4($sp)  add $fp,$sp,$0  subi $sp,$sp,32  lw $t1,4($fp)  li $t2,1  bne $t1,$t2,L1  li $v1,1  #return  add $sp,$fp,$0  lw $ra,0($sp)  addi $sp,$sp,4  lw $fp,-8($sp)  jr $ra  j L2  L1:  L2:  lw $t1,4($fp)  li $t2,2  bne $t1,$t2,L3  li $v1,1  #return  add $sp,$fp,$0  lw $ra,0($sp)  addi $sp,$sp,4  lw $fp,-8($sp)  jr $ra  j L4  L3:  lw $t1,4($fp)  li $t2,1  sub $t3,$t1,$t2  sw $t3,-16($fp)  subi $sp,$sp,4  lw $t1,-16($fp)  sw $t1,0($sp)  subi $sp,$sp,4  jal Fibon1  addi $sp,$sp,4  sw $v1,-20($fp)  lw $t1,4($fp)  li $t2,2  sub $t3,$t1,$t2  sw $t3,-24($fp)  subi $sp,$sp,4  lw $t1,-24($fp)  sw $t1,0($sp)  subi $sp,$sp,4  jal Fibon1  addi $sp,$sp,4  sw $v1,-28($fp)  lw $t1,-20($fp)  lw $t2,-28($fp)  add $t3,$t1,$t2  sw $t3,-32($fp)  lw $v1,-32($fp)  #return  add $sp,$fp,$0  lw $ra,0($sp)  addi $sp,$sp,4  lw $fp,-8($sp)  jr $ra  L4:  main:  #main head  sw $ra,($sp)  add $fp,$sp,$0  subi $sp,$sp,8  li $t1,6  sw $t1,-8($fp)  subi $sp,$sp,4  li $t1,6  sw $t1,0($sp)  subi $sp,$sp,4  jal Fibon1  addi $sp,$sp,4  sw $v1,-12($fp)  lw $t1,-12($fp)  la $t2,ret  sw $t1,0($t2)  la $t1,ret  lw $a0,0($t1)  li $v0,1  syscall  #return main |

### 2.2.3运行结果：

输入参数为5，五项斐波那契数列之后。

Mars软件的输出：



测试通过

## 2.2冒泡排序

### 2.2.1测试代码

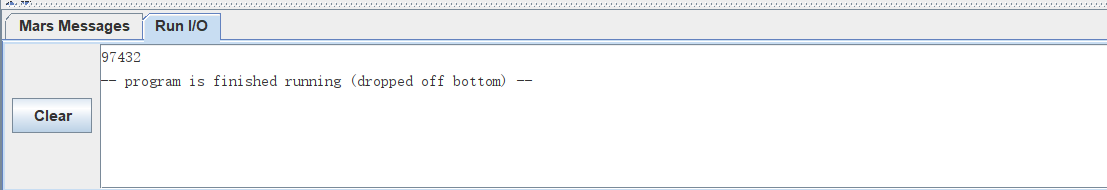
|  |
| --- |
| int main()  {  int a[5];  int i;  int j;  int t;  a[0]=9;  a[1]=3;  a[2]=4;  a[3]=2;  a[4]=7;  i=0;  j=0;  while(i<4)  {  j=0;  while(j<4)  {  if(a[j]<a[j+1])  {  t=a[j];  a[j]=a[j+1];  a[j+1]=t;  }  j=j+1;  }  i=i+1;  }  i=0;  putint(a[0]);  putint(a[1]);  putint(a[2]);  putint(a[3]);  putint(a[4]);  return 0;    } |

### 2.2.2 汇编代码

|  |
| --- |
| .data  a: .space 20  .text  j main  main:  #main head  sw $ra,($sp)  add $fp,$sp,$0  subi $sp,$sp,80  li $t1,0  sll $t1,$t1,2  la $t2,a  add $t3,$t2,$t1  li $t2,9  sw $t2,0($t3)  li $t1,1  sll $t1,$t1,2  la $t2,a  add $t3,$t2,$t1  li $t2,3  sw $t2,0($t3)  li $t1,2  sll $t1,$t1,2  la $t2,a  add $t3,$t2,$t1  li $t2,4  sw $t2,0($t3)  li $t1,3  sll $t1,$t1,2  la $t2,a  add $t3,$t2,$t1  li $t2,2  sw $t2,0($t3)  li $t1,4  sll $t1,$t1,2  la $t2,a  add $t3,$t2,$t1  li $t2,7  sw $t2,0($t3)  li $t1,0  sw $t1,-8($fp)  lw $t1,-8($fp)  sw $t1,-12($fp)  li $t1,0  li $t2,4  bge $t1,$t2,L1  L2:  li $t1,0  sw $t1,-12($fp)  li $t1,0  li $t2,4  bge $t1,$t2,L3  L4:  lw $t1,-12($fp)  sll $t1,$t1,2  la $t2,a  add $t3,$t2,$t1  lw $t2,0($t3)  sw $t2,-28($fp)  lw $t1,-12($fp)  li $t2,1  add $t3,$t1,$t2  sw $t3,-32($fp)  lw $t1,-32($fp)  sll $t1,$t1,2  la $t2,a  add $t3,$t2,$t1  lw $t2,0($t3)  sw $t2,-36($fp)  lw $t1,-28($fp)  lw $t2,-36($fp)  bge $t1,$t2,L5  lw $t1,-12($fp)  sll $t1,$t1,2  la $t2,a  add $t3,$t2,$t1  lw $t2,0($t3)  sw $t2,-44($fp)  lw $t1,-44($fp)  sw $t1,-16($fp)  lw $t1,-12($fp)  li $t2,1  add $t3,$t1,$t2  sw $t3,-48($fp)  lw $t1,-48($fp)  sll $t1,$t1,2  la $t2,a  add $t3,$t2,$t1  lw $t2,0($t3)  sw $t2,-52($fp)  lw $t1,-12($fp)  sll $t1,$t1,2  la $t2,a  add $t3,$t2,$t1  lw $t2,-52($fp)  sw $t2,0($t3)  lw $t1,-48($fp)  sw $t1,-56($fp)  lw $t1,-56($fp)  sll $t1,$t1,2  la $t2,a  add $t3,$t2,$t1  lw $t2,-16($fp)  sw $t2,0($t3)  j L6  L5:  L6:  lw $t1,-12($fp)  li $t2,1  add $t3,$t1,$t2  sw $t3,-12($fp)  lw $t1,-12($fp)  li $t2,4  blt $t1,$t2,L4  L3:  lw $t1,-8($fp)  li $t2,1  add $t3,$t1,$t2  sw $t3,-8($fp)  lw $t1,-8($fp)  li $t2,4  blt $t1,$t2,L2  L1:  li $t1,0  sw $t1,-8($fp)  li $t1,0  sll $t1,$t1,2  la $t2,a  add $t3,$t2,$t1  lw $t2,0($t3)  sw $t2,-68($fp)  lw $a0,-68($fp)  li $v0,1  syscall  li $t1,1  sll $t1,$t1,2  la $t2,a  add $t3,$t2,$t1  lw $t2,0($t3)  sw $t2,-72($fp)  lw $a0,-72($fp)  li $v0,1  syscall  li $t1,2  sll $t1,$t1,2  la $t2,a  add $t3,$t2,$t1  lw $t2,0($t3)  sw $t2,-76($fp)  lw $a0,-76($fp)  li $v0,1  syscall  li $t1,3  sll $t1,$t1,2  la $t2,a  add $t3,$t2,$t1  lw $t2,0($t3)  sw $t2,-80($fp)  lw $a0,-80($fp)  li $v0,1  syscall  li $t1,4  sll $t1,$t1,2  la $t2,a  add $t3,$t2,$t1  lw $t2,0($t3)  sw $t2,-84($fp)  lw $a0,-84($fp)  li $v0,1  syscall  #return main |

### 2.2.3运行结果

对数组进行升序排序



# sysy测试用例

没有正确通过编译。原因有：

没有实现取模运算。

没有对数组的维度进行正确处理（编译时可以求出的常量也可以作为数组的大小）

没有使用标准的sysy运行时库进行IO处理

函数参数无法传递数组

条件表达式有多个条件的情况没有实现

# 总结

通过编写测试用例，找出不同模块的bug，对sysy的语义约束以及最后代码的正确新进行检查。

测试用例分为两部分，一个部分是自己编写的测试用例，用于检查实现的功能，并对错误的地方进行debug修改。另一部分是2020年编译器设计决撒中sysy的测试用例，因为sysy规定的文法有地方没有实现，因此没有通过比赛的测试用例。