



西安交通大学
XI'AN JIAOTONG UNIVERSITY

2022-2023 学年第二学期

《编译器设计专题实验》

实验报告 1

学 院： 电信学部
班 级： [REDACTED]
学 号： [REDACTED]
姓 名： [REDACTED]

二〇二三年 四月

目录

一、实验内容（必做 1）	1
二、实验内容（必做 2）	2
三、实验结果.....	2
(1) 修改终端提示符	2
(2) 编译 Hello World 程序.....	3
(3) 词法分析	3
(4) 语法分析	4
(5) 语义分析	4
(6) 代码生成	5
(7) 程序运行	5
(8) Cool 语言程序-从 1 到 100 求和	5
(9) Cool 语言程序-求圆的面积	5
四、源代码.....	6
(1) Hello World.....	6
(2) 从 1 到 100 求和	6
(3) 计算圆的面积	7

《实验 1-环境配置和 HelloWorld》

一、实验内容（必做 1）

1. 修改终端提示符，让代码和界面成为唯一标识符；
2. 标准的 Cool 语言编译器 Coolc 在 bin 目录下，进入该目录后，使用如下指令编译运行 hello_world.cl 程序：

```
①./coolc ../examples/hello_world.cl
```

```
②./spim -trap_file ../lib/trap.handler -file ../examples/hello_world.s
```

3. 用标准的词法分析程序，编译 Cool 语言程序，观察结果，命令为：

```
./reference-lexer ../examples/hello_world.cl
```

4. 用标准的词法分析程序和语法分析程序，编译 Cool 语言程序，观察结果，命令为：

```
./reference-lexer ../examples/hello_world.cl | ./reference-parser
```

其中，'|' 为 Linux 下的管道流，意思是将前一个命令的输出作为后一个命令的输入。

5. 用标准的词法分析程序、语法分析程序和语义分析程序编译 Cool 语言程序，观察结果，命令为：

```
./reference-lexer ../examples/hello_world.cl | ./reference-parser  
| ./reference-semant
```

6. 用标准的词法分析程序、语法分析程序、语义分析程序和代码生成程序共同编译 Cool 语言程序，生成最终的汇编代码，观察结果，命令为：

```
./reference-lexer ../examples/hello_world.cl | ./reference-parser  
| ./reference-semant | ./reference-cgen
```

7. 使用输出重定向符 '>'，将得到的汇编代码输出到.s 文件中，命令为：

```
./reference-lexer ../examples/hello_world.cl | ./reference-parser  
| ./reference-semantic | ./reference-cgen > code.s
```

8. 将生成的汇编代码在 spim 上运行，命令为：

```
./spim -trap_file ../lib/trap.handler -file code.s
```

二、实验内容（必做 2）

编写简单的 Cool 语言程序，并按照前面说的方法编译并执行程序。

三、实验结果

（1）修改终端提示符

①方法一：

PPT 中并无给出具体的操作过程，但根据指令判断应该是通过 `sudo -i` 指令以 root 用户身份打开一个新的登录 shell，再通过 vim 修改 Ubuntu 的默认 `bashrc` 文件来修改终端提示符，但这样做只修改了新建用户的配置文件，退出后仍会恢复原来的命令提示符，且 vim 编辑较为麻烦。

可以通过以下方法修改命令提示符，且下次重新登录后命令提示符保持不变：

```
guo@guo-virtual-machine:~$ gedit ~/.bashrc  
guo@guo-virtual-machine:~$ source ~/.bashrc  
GuoSongjian(Thu Apr 13 20:10:44):~$
```

通过 `gedit` 指令可以在文本编辑器的图形化界面中修改 `bashrc` 文件，在文件最后一行增加如下内容：

```
export PS1="\[$(tput setaf 2)\]GuoSongjian(\d \t):\w$ \[$(tput sgr0)\]"
```

其中，`'\[$(tput tsetaf 2)\]'` 用于设置颜色为绿色，`'\[$(tput sgr0)\]'` 用于重置颜色，`'\d'` 表示日期，`'\t'` 表示时间，`'\w'` 表示当前工作目录。

最后，执行 source 指令让新的配置文件生效。

②方法二：

通过如下指令直接修改命令提示符：

```
guo@guo-virtual-machine:~$ PS1="\[$(tput setaf 2)\]GuoSongjian(\d \t):~\w$ \[$(tput sgr0)\]"
GuoSongjian(Thu Apr 13 20:31:58):~$
```

缺点是打开新的终端后命令提示符会恢复原样。

(2) 编译 Hello World 程序

```
GuoSongjian(Thu Apr 13 20:38:08):~/compiler_exp/cool/cool/bin$ ./coolc ../examples/hello_world.cl
bash: ./coolc: Permission denied
GuoSongjian(Thu Apr 13 20:38:57):~/compiler_exp/cool/cool/bin$ sudo ./coolc ../examples/hello_world.cl
[sudo] password for guo:
sudo: ./coolc: command not found
GuoSongjian(Thu Apr 13 20:39:11):~/compiler_exp/cool/cool/bin$ chmod a+x coolc
GuoSongjian(Thu Apr 13 20:39:28):~/compiler_exp/cool/cool/bin$ sudo ./coolc ../examples/hello_world.cl
GuoSongjian(Thu Apr 13 20:39:32):~/compiler_exp/cool/cool/bin$ sudo ./spin -trap_file ../lib/trap.handler -file ../examples/hello_world.s
sudo: ./spin: command not found
GuoSongjian(Thu Apr 13 20:41:36):~/compiler_exp/cool/cool/bin$ chmod a+x spin
GuoSongjian(Thu Apr 13 20:41:44):~/compiler_exp/cool/cool/bin$ sudo ./spin -trap_file ../lib/trap.handler -file ../examples/hello_world.s
SPIM Version 6.5 of January 4, 2003
Copyright 1990-2003 by James R. Larus (larus@cs.wisc.edu).
All Rights Reserved.
See the file README for a full copyright notice.
Loaded: ../lib/trap.handler
Hello, World.
COOL program successfully executed
GuoSongjian(Thu Apr 13 20:41:46):~/compiler_exp/cool/cool/bin$
```

按照 PPT 上的步骤输入指令后，报错 Permission denied，于是使用 sudo 命令以管理员身份运行，但仍报错 command not found，查阅资料后发现是由于通过拷贝直接上传至 Linux 环境下的可执行文件没有执行权限，因此需要使用 chmod 指令给可执行文件添加执行权限。重新输入指令后程序正常执行。

(3) 词法分析

```
GuoSongjian(Thu Apr 13 20:57:42):~/compiler_exp/cool/cool/bin$ ./reference-lexer ../examples/hello_world.cl
#name "../examples/hello_world.cl"
#1 CLASS
#1 TYPEID Main
#1 INHERITS
#1 TYPEID IO
#1 '{'
#2 OBJECTID main
#2 '('
#2 ')'
#2 ':'
#2 TYPEID SELF_TYPE
#2 '{'
#3 OBJECTID out_string
#3 '('
#3 STR_CONST "Hello, World.\n"
#3 ')'
#4 '}'
#4 ';'
#5 '}'
#5 ';'

```

输出中的每一行表示一个词法单元，并由‘#’字符标记所处行号。对于 CLASS、INHERITS、括号、冒号、分号等关键字并不作标

记，而类 Main、IO 标记为 TYPEID，类中的方法 main、out_string 标记为 OBJECTID，字符串常量"Hello, World.\n"标记为 STR_CONST。

(4) 语法分析

```
GuoSongjian(Thu Apr 13 20:59:49):~/compiler_exp/cool/cool/bin$ ./reference-lexer ../examples/hello_world.cl |
./reference-parser
#5
_program
#5
_class
Main
IO
"../examples/hello_world.cl"
(
#4
_method
main
SELF_TYPE
#3
_dispatch
#3
_object
self
: _no_type
out_string
(
#3
_string
"Hello, World.\n"
: _no_type
)
: _no_type
)
```

语法分析检查词法分析器输出的单词序列是否是源语言中的句子，亦即是否符合源语言的语法规则。

(5) 语义分析

```
GuoSongjian(Thu Apr 13 21:03:49):~/compiler_exp/cool/cool/bin$ ./reference-lexer ../examples/hello_world.cl |
./reference-parser| ./reference-semantic
#5
_program
#5
_class
Main
IO
"../examples/hello_world.cl"
(
#4
_method
main
SELF_TYPE
#3
_dispatch
#3
_object
self
: SELF_TYPE
out_string
(
#3
_string
"Hello, World.\n"
: String
)
: SELF_TYPE
)
```

语义分析负责审查源程序有无语义错误，为代码生成阶段收集类型信息。

(6) 代码生成

```
GuoSongjian(Thu Apr 13 21:09:41):~/compiler_exp/cool/cool/bin$ ./reference-lexer ../examples/hello_world.cl |
./reference-parser | ./reference-semant | ./reference-cgen
.data
.align 2
.globl class_nameTab
.globl Main_protObj
.globl Int_protObj
.globl String_protObj
.globl bool_const0
.globl bool_const1
.globl _int_tag
.globl _bool_tag
.globl _string_tag
_int_tag:
.word 3
_bool_tag:
.word 4
_string_tag:
.word 5
.globl _MemMgr_INITIALIZER
_MemMgr_INITIALIZER:
.word _NoGC_Init
.globl _MemMgr_COLLECTOR
_MemMgr_COLLECTOR:
.word _NoGC_Collect
.globl _MemMgr_TEST
_MemMgr_TEST:
.word 0
.word -1
str_const9:
.word 5
.word 5
.word String_dispTab
```

将 Cool 语言的源程序转换为 MIPS 汇编代码。

(7) 程序运行

```
GuoSongjian(Thu Apr 13 21:10:08):~/compiler_exp/cool/cool/bin$ ./reference-lexer ../examples/hello_world.cl
| ./reference-parser | ./reference-semant | ./reference-cgen > code.s
GuoSongjian(Thu Apr 13 21:12:42):~/compiler_exp/cool/cool/bin$ ./spim -trap_file ../lib/trap.handler -file
code.s
SPIM Version 6.5 of January 4, 2003
Copyright 1990-2003 by James R. Larus (larus@cs.wisc.edu).
All Rights Reserved.
See the file README for a full copyright notice.
Loaded: ../lib/trap.handler
Hello, World.
COOL program successfully executed
```

程序运行结果与步骤（2）相同。

(8) Cool 语言程序-从 1 到 100 求和

```
GuoSongjian(Thu Apr 13 23:12:28):~/compiler_exp/cool/cool/bin$ sudo ./coolc ../../exp1/sum.cl
GuoSongjian(Thu Apr 13 23:13:00):~/compiler_exp/cool/cool/bin$ sudo ./spim -trap_file ../lib/trap.handler -file ../../exp1/sum.s
SPIM Version 6.5 of January 4, 2003
Copyright 1990-2003 by James R. Larus (larus@cs.wisc.edu).
All Rights Reserved.
See the file README for a full copyright notice.
Loaded: ../lib/trap.handler
sum from 1 to 100: 5050
COOL program successfully executed
```

需要注意，Cool 语言中分号是表达式的终结符而非分隔符。

(9) Cool 语言程序-求圆的面积

```
GuoSongjian(Thu Apr 13 23:38:36):~/compiler_exp/cool/cool/bin$ sudo ./coolc ../../exp1/circle.cl
GuoSongjian(Thu Apr 13 23:39:18):~/compiler_exp/cool/cool/bin$ sudo ./spim -trap_file ../lib/trap.handler -file ../../exp1/circle.s
SPIM Version 6.5 of January 4, 2003
Copyright 1990-2003 by James R. Larus (larus@cs.wisc.edu).
All Rights Reserved.
See the file README for a full copyright notice.
Loaded: ../lib/trap.handler
input radius: 3
28.26
COOL program successfully executed
```

由于 Cool 语言中无浮点类型，这里计算圆面积的处理方法是将圆周率取 314 后再参与运算，输出结果时再将整数分割并定位输出小数点字符。

四、源代码

(1) Hello World

```
class Main inherits IO {  
    main(): SELF_TYPE {  
        out_string("Hello, World.\n")  
    };  
};
```

(2) 从 1 到 100 求和

```
class Main inherits IO {  
    sum: Int <- 0;  
    i: Int <- 1;  
    main(): SELF_TYPE { {  
        while i <= 100 loop {  
            sum <- sum + i;  
            i <- i + 1;  
        } pool;  
        out_string("sum from 1 to 100: ");  
        out_int(sum);  
        out_string("\n");  
    } };  
};
```


(3) 计算圆的面积

```
class Main inherits IO {  
  main(): SELF_TYPE { {  
    out_string("input radius: ");  
    let r:Int <- in_int(), area:Int in {  
      area <- 314 * r * r;  
      out_int(area / 100);  
      out_string(".");  
      out_int(area - area / 100 * 100);  
      out_string("\n");  
    };  
  } };  
};
```