

Lab. 1 初代编译器实验说明和要求

一、初代编译器功能描述

初代编译器将 C 语言顺序语句序列翻译为等价的汇编程序，所输出的汇编程序符合 x86 或 MIPS 汇编语言格式要求，能够被后续的汇编器翻译为可执行程序运行。如果目标平台为 x86，则生成的可执行程序能够在目标机上执行和验证结果；如果目标平台为 MIPS，则生成的汇编程序可以在 MIPS 模拟器中运行和验证结果。

二、初代编译器文法要求

初代编译器能够处理的文法如下所示：

关键字： int, return

标识符： 单个英文字母，如 a、b 等

常量： 十进制整型，如 1、223、10 等

操作符： =、+、-、*、/

分隔符： ；

语句： 表达式语句、赋值语句

三、初代编译器测试样例

测试样例不用考虑优先级，即表达式中算术操作符优先级相同，且无括号。

所有输入测试样例文件中单词之间均由空格或者回车分隔，但输入文件中可能存在多个连续的空格或者回车。

评分依据 return 的值是否符合预期。生成的 MIPS 汇编代码在 QtSpim 上运行。

输入样例：

```
int a;  
int b;  
int d;  
a = 1;  
b = 2;  
d = a + b;  
return d;
```

（预期返回值为 3）

输出样例 x86:

```
.intel_syntax noprefix
.global main
.type main, @function
main:
    push    ebp
    mov     ebp,esp
    sub     esp,16
    mov     eax,1
    mov     DWORD PTR [ebp-4],eax
    mov     eax,2
    mov     DWORD PTR [ebp-8],eax
    mov     eax,DWORD PTR [ebp-4]
    add     eax,DWORD PTR [ebp-8]
    mov     DWORD PTR [ebp-12],eax
    mov     eax,DWORD PTR [ebp-12]
    leave
    ret
```

输出样例 mips:

```
.globl main
main:
    addiu $sp,$sp,-32
    sw $fp,28($sp)
    move $fp,$sp
    li $2,1 # 0x1
    sw $2,8($fp)
    li $2,2 # 0x2
    sw $2,12($fp)
    lw $3,8($fp)
    lw $2,12($fp)
    nop
    addu $2,$3,$2
    sw $2,16($fp)
    lw $2,16($fp)
    move $sp,$fp
    lw $fp,28($sp)
    addiu $sp,$sp,32
    jr $31
    nop
```

四、初代编译器实现参考

初代编译器的实现基于程序设计基础、算法和数据结构等课程所学知识。在词法分析部分可以使用正则匹配或其他思路实现，而代码生成部分则可以使用栈来完成。

其中词法分析部分的实现思路是：根据空格或者回车将输入源码字符串分割为多个子串，然后判断每个子串属于哪个单词类，整型常量按照对应规则匹配或最长数字串匹配，连续字母考虑匹配 `return`，单个字母即标识符，其他单词直接与目标字符串比较匹配即可。

其中代码生成部分的参考实现思路如下：

1. 变量声明语句

对于变量声明，需要在栈帧中预先留出相关空间。

x86

```
int a;  
int b;
```

对应的汇编是

```
sub esp,8           #预留两个 4 字节空间  
mov DWORD PTR [ebp-4],0 #第 1 个变量（地址为 DWORD PTR [ebp-4]）赋值 0  
mov DWORD PTR [ebp-8],0 #第 2 个变量（地址为 DWORD PTR [ebp-8]）赋值 0
```

MIPS

与 x86 大致相同，例如，在初始化栈帧时 `addiu $sp,$sp,-32`，调整栈指针（`$sp`），为局部变量和保存的寄存器分配 32 字节的空间

2. return 语句

一个文件只含有一个 `return` 语句，遇到 `return` 语句时从 `main` 函数返回。

x86

将返回值复制到 `eax` 寄存器中，例如 `return d;` 对应 `mov eax, DWORD PTR [ebp-12]`

MIPS

将返回值复制到 `$v0`（即 `$2`）寄存器中。

3. 赋值与表达式语句

不区分运算符优先级的话，按照左结合顺序依次入栈出栈计算即可。

具体指令请参考对应汇编语言手册。

注意：“三”中的样例输出以及“四”中代码生成部分的参考思路中使用了 `sp`，可以考虑使用简单的设计思路，考虑变量与寄存器一一映射，直接使用寄存器，不需要使用 `sp`。

五、初代编译器提交要求

实现语言：C++（语言标准 `c++14`）

编译环境：g++

提交内容：单个 `cpp` 源文件，文件名称为 `compilerlab1.cpp`

输入输出：实现的编译器有一个命令行参数，用于指明输入文件路径，编译器从该路径读取源码，并向输出编译结果到以 `.s` 结尾的汇编代码文件中。

六、初代编译器参考文档附件

1、MIPS 汇编

参考网址：<https://freeflyingsheep.github.io/posts/mips/assembly/>

2、SPIM Simulator

生成的 MIPS 指令可以在模拟器上运行，有两个版本：

（1）命令行版

安装：`sudo apt-get install spim`

测试：`spim -file [汇编代码文件名]`

（2）GUI 版（可以单步调试）

下载地址：<http://pages.cs.wisc.edu/~larus/spim.html>

使用方法：载入程序生成的目标代码文件

MIPS 和 SPIM Simulator 的内容还可以参考《编译原理实践与指导教程》相关部分。