

编译原理实验

徐伟元 16340261

Mail: xuwy27@mail2.sysu.edu.cn

Major: Software Engineering

School: School of Data and Computer Science

University: Sun Yat-Sen University

2018 年 12 月 24 日

1 实验内容

1. 编译运行 PASCAL 语言的 PL0 编译程序;
2. PL0 的编译程序运行时, 通过输入文件输入 PL0 源程序, 在输出文件中产生源程序的中间代码, 然后运行该中间代码, 在输出文件中产生运行数据;
3. 在 PL0 语言中增加 Read 和 Write 语句, 使得 PL0 源程序可以使用 Read 和 Write 语句, 从文件 (或键盘) 输入数据, 并可以向文件 (或屏幕) 写数据.

2 编译运行 PL0 编译程序

2.1 修改 PL0 编译程序

使用 Free Pascal 在 windows 上构建 Pascal 环境后, 修改 PL0 编译程序。编译程序主要的问题有如下几个:

1. 关系符错乱
2. 使用保留字作为变量名
3. 程序中使用中文符号

编译程序使用手写数学符号作为关系符, 而在 PASCAL 语言中, 这些符号被认为是字符串类型, 所以更改符号识别过程, 使其识别 $<>$, $<=$, $>=$ 关系符。其使用 Procedure 关键字作为变量名, 更改为非保留字即可。至于中文符号, 直接替换, 主要有中文破折号作为减号和中文引号。更换完毕后, 编译通过。

2.2 文件读写

使用 PASCAL 的文件读写完成编译程序的文件读写功能, 即使用 reset 打开文件, 使用 rewrite 写入文件。

通过修改编译程序的运行代码, 可以完成不同数据的输出 (见附件):

1. PL0 编译程序-1

2. PL0 源程序-1
3. 中间代码-1
4. 栈中数据-1

3 拓展 PL0 语言

3.1 增加 read 和 write 语句

这里我们修改 PL0 编译程序，增加保留关键字数目，函数代码和符号码。

在符号识别时，如果读取到 read 关键字，读入'(', 如无，则报错，然后持续读入数据，若读入非符号表内变量，报错；直到读入')'，产生中间代码，然后判断读入符号合法性。在运行中间代码执行 read 函数时，利用 PASCAL 的 read 函数往栈中读入数据，即可。

在符号识别时，如果读取到 write 关键字，读入'(', 如无，则报错，然后持续读入数据，若读入非符号表内变量，报错；直到读入')'，产生中间代码，然后判断读入符号合法性。在运行中间代码执行 write 函数时，利用 PASCAL 的 write 函数从栈中写出数据，即可。

3.2 修改 PL0 源程序

最后修改 PL0 的源程序。首先使用原来的乘法操作来测试输出语句，可以从图一和附件 (Output) 中看出，输出 7, 85 和 595。结果正常。然后增加读入语句，使用除法和最大公因数函数来测试 read 和 write 语句。可以从图一和附件 (Output) 中看出，输入的 25, 5 进行除法输出 5；输入的 99 和 33 进行求取最大公因数输出 33。结果均正常。

最终各项数据输出见附件：

1. PL0 编译程序-2
2. PL0 源程序-2
3. PL0 源程序运行结果-2
4. 中间代码-2
5. 栈中数据-2

```
PS C:\Users\43787\Desktop\pascal\compilers\Experiment2> fpc .\exp2.pas
Free Pascal Compiler version 3.0.4 [2017/10/06] for i386
Copyright (c) 1993-2017 by Florian Klaempfl and others
Target OS: Win32 for i386
Compiling .\exp2.pas
Linking exp2.exe
831 lines compiled, 0.1 sec, 38064 bytes code, 1268 bytes data
PS C:\Users\43787\Desktop\pascal\compilers\Experiment2> .\exp2.exe
please input source program file name : input
please input result file name : output
?:25
?:5
?:99
?:33
PS C:\Users\43787\Desktop\pascal\compilers\Experiment2> █
```

图 1: TerminalInput