

面向对象程序设计基础

大作业(1)汇编语言解释器

清华大学计算机科学与技术系 计05班

2010011351 皮一凡

piyifan@qq.com

1 概览

按面向对象的方法设计了一个汇编语言解释器，特性为：

- 支持最多26个('a'到'z')16位数据寄存器。本实现中支持(a-d)x: (a-d)h-(a-d)l。
- 支持标志寄存器，本实现中支持ZF与SF。
- 支持MOV、ADD、SUB、INC、DEC、INT操作命令。
- 支持标号与LOOP、JMP、JZ、JE、JNZ、JNE、JL、JG跳转命令。
- 能够对所解释汇编代码查错。
- 使用面向对象方式设计寄存器和语句，方便扩展。

使用说明：

- 使用方法为：asm.exe asm_source_file即可解释执行asm_source_file文件，如果该文件的代码出现错误，将报出错误位置。否则将解释执行该代码。
- 本解释器对大小写敏感，语句与寄存器字母必须小写，16进制数字字母需大写，否则可能报错。
- 允许行头和行末的多余空格和注释语句(；开头)。
- 输入输出的所有数字均为16进制，10-15依次用A-F表示。

2 设计思路

设计了asm.h、regisiter.h、label.h、command.h四种类型的类，说明如下：

- asm.h中包含Asm类。
Asm类为解释器类，能够读取源代码并检查编译错误，之后调用其他各类执行源代码。
- regisiter.h中包含Register、RegisterList、Flag、FlagList四个基本类以及SF、ZF两个标志寄存器类。
 - RegisterList为数据存储器列表，提供对某个数据存储器运算、赋值、读取的操作接口，并有输出所有数据存储器的存储数值以及清空所有寄存器的接口。使用该类存放支持的数据寄存器能够方便语句对数据寄存器进行操作。

- Register为数据寄存器类，为16进制寄存器，有对高位、低位和整个寄存器进行运算和赋值的操作接口，并能返回当前所存储数值以及名称。另外，每次进行数据运算均支持修改标志寄存器。数据全部封装，命令仅能通过RegisterList来操作Register。
 - FlagList为标志寄存器列表，提供读取某个标志寄存器、根据运算结果更新所有标志寄存器的接口。使用该类存放支持的标志寄存器能够方便语句对数据寄存器进行操作。
 - Flag为标志寄存器基类，所有标志存储器(本实现中有SF和ZF)继承自Flag，存储标志寄存器的名字与数值。并使用虚函数方便地进行不同类型标志寄存器的不同更新工作。
- label.h中包含Label和LabelList两个类。
 - LabelList为标号列表类，用于存储跳转命令所需的标号。其中使用vector存取Label类的对象，提供查找、新建和清空标号的操作接口。使用该类存放标号能够方便存储和查找标号。
 - Label为标号类，将标号的名称和位置封装，提供返回名字和位置的接口函数。
 - command.h中包含Command和CommandList两个基本类和14个具体命令类。
 - CommandList为命令列表，存储所有支持的命令。使用虚函数方便地实现对某语句查错和执行某语句的操作接口，同时执行命令接口返回一个整数，为0则表示执行完后不跳转，大于0则跳转自相应语句。使用该类存放支持的命令能利用多态性方便的实现对某个语句的查错和执行。
 - Command为命令基类，所有具体命令均继承自该类。提供返回名字、执行和查错操作接口。

另外，general.h提供了10、16进制互相转换和判断字符串是否为16进制的函数。main.cpp调用Asm类进行代码输入和解释执行工作。

3 扩展方式

由于使用面向对象方式进行设计，该解释器的代码方便扩展。具体如下：

- 增加新的存储寄存器：修改regisiter.h中的regs数组大小与regisiter.cpp中的数组定义和RegisiterList的构建，暂仅能添加名字为单个小写字母的寄存器。
- 增加新的标志寄存器：修改regisiter.h中flgs数组大小与regisiter.cpp中的数组定义和FlagList的构建。同时加入继承自Flag基类的标志寄存器类，按格式写出构建及更新函数即可。
- 增加新的命令：修改command.h中cmds数组大小与command.cpp中的数组定义和CommandList的构建。同时加入继承自Command基类的命令类，按格式写出构建、运行及查错函数即可。

4 附录

使用MinGW32 3.4.5在Windows 7环境下编译通过，并能正确的解释执行Ex文件夹中的汇编代码，输出预期结果。