Small 编译后的中间语言的指令集

|  |  |
| --- | --- |
| 指令码 | 说明 |
| push | 将一个整数或一个标识符压入运行栈 |
| pop | 将一个整数或一个标识符从运行栈内弹出，若弹出的是标识符，则标识符在符号表中的值被更新 |
| jz | 根据flag置位与否决定是否跳转 |
| jmp | 无条件跳转到指定标号处 |
| out | 将栈顶元素输出到显示终端 |
| in | 由用户输入标识符的值，将标识符在符号表中的值更新 |
| neg | 栈顶元素取相反数 |
| add | 将栈顶的两个整数相加后放入栈顶，原来的两个数被弹出 |
| sub | 栈顶两整数相减 |
| mul | 栈顶两整数相乘 |
| div | 栈顶两整数相除 |
| compLT | 栈顶两整数相比较，若底部元素小于顶部元素则flag=1 |
| compGT | 栈顶两整数相比较，若底部元素大于顶部元素则flag=1 |
| compLE | 栈顶两整数相比较，若底部元素小于等于顶部元素则flag=1 |
| compGE | 栈顶两整数相比较，若底部元素大于等于顶部元素则flag=1 |
| compEQ | 栈顶两整数相比较，若底部元素等于顶部元素则flag=1 |
| compNE | 栈顶两整数相比较，若底部元素不等于顶部元素则flag=1 |
| halt | 停止程序执行，不再执行后续语句 |
| stm | 将数存储入数组名所指向的地址加上偏移量的位置 |
| ldm | 将数组名所指向的地址加上偏移量的位置的值取出 |
| new | 说明该标识符是一个数组的首地址 |
| alloc | 必须紧跟在new之后，为数组开辟空间 |
| and | 仅跟在comp系列指令之后，将栈顶两数做逻辑与运算，改变控制寄存器的值 |
| or | 同上，将栈顶两数做逻辑或运算 |
| not | 同上，将栈顶数做逻辑非运算（非零值变成0，0变成1） |
| bitand | 将栈顶两数做位与运算 |
| bitor | 将栈顶两数做位或运算 |
| bitnot | 将栈顶数做取反运算 |
| Bitxor | 将栈顶两数做异或运算 |
| inc | 栈顶元素作自增运算 |
| dec | 栈顶元素作自减运算 |