微机原理实验四

PB13206106
罗勇冠

一.实验目的

通过本次实验熟练掌握汇编语言的字符读取、字符处理和字符输出操作,并学会用汇编语言计算整数加减运算表达式。

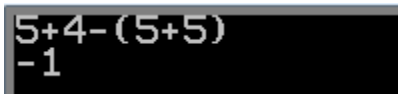
二.实验内容

题 4：编程计算任一整数加减运算表达式。其中，表达式从键盘输入，可带括号；操作数为字数据。表达式的长度不超过 1024 个字节。

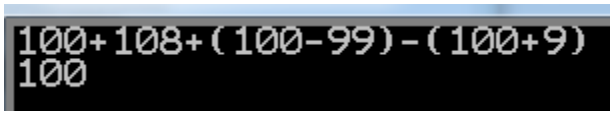
三.实验结果

测试 1：1-(4-5) ,输出结果为一位数正数
结果：2

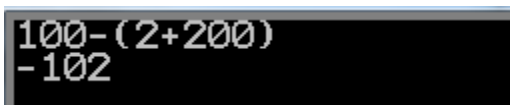
测试 2：5+4-(5+5) ,输出结果为一位数负数
结果：-1



测试 3：100+108+(100-99)-(100+9)
结果：100 ,输出结果为多位数正数



测试 4：100-(2+200)
结果：-102 ,输出结果为多位数负数



四.实验分析

实验整体思路是建立三个数组，第一个数组 BUF 读取表达式，包含’[0-9]’、’(’、’)’、’+’、’-’，回车键结束输入，表达式的长度不超过 1024。再将连续的单个数字转化为多位数，表达式不变，存入第二个数组 ARRAY1。对第二个数组进行处理，从后往前读，若读到的为’(’，将其标记为’$’，并从该位置往后读，寻找第一个配对的’)’，同样将其标记为’$’。判断之前作标记的’(’前面的字符，若为’-’，表示配对的括号内所有运算符要改变符号，即’+’变’-’，’-’变’+’。当处理到第二个数组开始位置时，处理结束。继续扫描，将所有标号为’$’代表括号的字符忽略，

剩余数字和运算符存入第三个数组 ARRAY2 中，处理结束以后，便可以对第三个数组中的内容进行运算操作。读取第一个数字，判断下一个运算符若为’+’，则与运算符后的数字进行相加操作，同理，若下一个运算符为’-’，则与运算符后的数字进行相减操作，直到处理完所有元素，最终结果存放至 AX 中。至于输出部分，先输出回车换行符，考虑结果的正负性，判断结果最高位是否为 0，若否，表示负数，结果先输出’-’，将负数取绝对值，再进行除法压栈输出。若为正数，直接进行除法压栈输出，进而打印出最终结果。

五.意见建议

感谢检查实验的助教给了我不错的建议，一开始处理数据的时候用了四个数组，额外的一个用来存放运算符，助教提示告诉说运算符数组可以用第二个数组代替，于是尝试删除了该数组，简化了代码。另外，一开始忽略了结果的正负，在助教的测试数据下，发现当结果为负数时，输出错误。后来的修改中，在输出部分，加上判断结果最高位是否为 1 的代码，如果是负数先输出’-’，再将负数取绝对值除 10 压栈输出。输出部分如下:

```
PRINT:
    MOV DI,0
    MOV BX,10
    TEST AX,8000H                ;判断最高位
    JZ P0                        ;正数
    PUSH AX                       ;负数，输出'-'

    MOV AX,0E0AH                  ;CRLF
    INT 10H
    MOV AX,0E0DH
    INT 10H

    MOV AX,0E2DH                  ;'-' ASCII
    INT 10H

    POP AX
    NEG AX
    MOV DI,1

    P0:
    MOV DX,0
    XOR CX,CX
    Q0:
    XOR DX,DX
    DIV BX
    XOR DX,0E30H
    PUSH DX
    INC CX
    CMP AX,0
    JNZ Q0
    CMP DI,1
```

```
    JE Q1

    MOV AX,0E0AH                ;CRLF
    INT 10H
    MOV AX,0E0DH
    INT 10H

    Q1:
    POP AX
    INT 10H
    LOOP Q1
    JMP ENDF
```

六.源代码

```
DATA SEGMENT
    BUF DB 1025 DUP (0)
    ARRAY2 DW 256 DUP (0)
    ARRAY1 DW 256 DUP (0)
    ERRORINFO DB 0AH,0DH, 'ERROR!', '$'
    RANGE DW 0
    NUMBER DW 0
DATA ENDS

CODE SEGMENT
    ASSUME CS:CODE, DS:DATA
    START:
    MOV AX, DATA
    MOV DS, AX

    MOV SI, 0
    INPUT:                  ;输入中只能有数字、+、-、(、)、回车
    MOV AH, 1
    INT 21H
    CMP AL, 0DH             ;若为回车，读取所有字符结束，跳转
    JE CON
    CMP AL, 28H             ;若是 0DH(回车)~28H('(')中间任意字符，跳转错误
    JB ERROR
    CMP AL, 2AH             ;若是 2AH('*')，跳转错误
    JE ERROR
    CMP AL, 2CH             ;若是 2CH(','),跳转错误
    JE ERROR
    CMP AL, 2EH             ;若是 2EH('.')，跳转错误
    JE ERROR
    CMP AL, 2FH             ;若是 2FH('/')，跳转错误
```

```
JE ERROR
CMP AL, 39H                ;若大于 39H('9')，跳转错误
JA ERROR
MOV BUF[SI], AL            ;将正确的字符放到 BUF 数组中
INC SI                     ;下标 SI 自增
CMP SI, 1024               ;若符号个数不超过最大限制 1024 且没按回车，继续读
JB INPUT

CON:                       ;读取结束标号
MOV RANGE, SI              ;记录总输入字符数
MOV SI, 0                  ;初始输入数组下标
MOV DI, 0                  ;初始录入数组下标
CHCKIN:
MOV BL, BUF[SI]
XOR BH, BH                 ;将输入字赋给 BX
CMP BUF[SI], 30H           ;判断是否为运算符
JB OPR                     ;小于 30H,是运算符，跳转
MOV AX, ARRAY2[DI]         ;不是运算符乘十录入
MOV CX, 10
MUL CX
MOV ARRAY2[DI], AX
SUB BL, 30H
ADD ARRAY2[DI], BX
CMP BUF[SI+1], 30H         ;输入数组下一个符号和 30H 比较
JB ADDON                   ;小于跳转
JMP FORW                   ;否则跳转
OPR: ;运算符处理
MOV ARRAY2[DI], BX         ;是运算符则直接录入

JMP ADDON
ADDON:
ADD DI, 2                  ;录入数组下标自增
FORW:
INC SI                     ;输入数组下标自增
CMP SI, RANGE              ;若未读取完所有符号，跳转回继续读
JB CHCKIN
MOV NUMBER, DI             ;若已读完所有符号，记录输入数组所有个数

MOV SI, DI
SEARCHL:
CMP SI, 0
JE READEND
SUB SI, 2
CMP ARRAY2[SI], '('
```

```
JNE SEARCHL                ;找到最右的一个左括号
MOV ARRAY2[SI], '$'
MOV DI, SI
FINDRHT:
ADD DI, 2
CMP ARRAY2[DI], ')'
JNE FINDRHT
MOV ARRAY2[DI], '$'          ;找到最近的一个右括号并标记
CMP SI, 0
JE SEARCHL
CMP ARRAY2[SI-2], '('
JE SEARCHL
CMP ARRAY2[SI-2], '+'
JE SEARCHL                 ;不改变加减号
CMP ARRAY2[SI-2], '-'
JE CHAN
CHAN:                       ;括号前为减号，改变括号内加减号
MOV BX, SI
REVERSE:
ADD BX, 2
CMP ARRAY2[BX], '+'
JE CHANADD
CMP ARRAY2[BX], '-'
JE CHANSUB
JMP SEERAN
CHANADD:                    ;改变加号
MOV ARRAY2[BX], '-'
JMP SEERAN
CHANSUB:                    ;改变减号
MOV ARRAY2[BX], '+'
JMP SEERAN
SEERAN:                     ;检验括号内所有运算符是否全部改变完
CMP BX, DI
JB REVERSE
JMP SEARCHL

READEND:
MOV SI, 0
MOV DI, 0
SCANIN:                     ;开始从头扫描
CMP SI, NUMBER
JE CALCULATE               ;全部扫描结束，进入计算阶段
CMP ARRAY2[SI], '$'
JNE SCANF
```

```
ADD SI, 2                    ;扫描发现配对括号，跳过括号
JMP SCANIN
SCANF:                       ;加减号，将 NUM 数组放入 LANUM 数组
MOV BX, ARRAY2[SI]
MOV ARRAY1[DI], BX           ;LANUM 数组中只含数字和加减号
ADD SI, 2
ADD DI, 2
JMP SCANIN

CALCULATE:
MOV SI, 0
MOV AX, ARRAY1[SI]
CALCU:
ADD SI, 2
CMP SI, DI
JAE PRINT
CMP ARRAY1[SI], 2BH          ;数字后面为加号，进入加法
JE ADDCAL
CMP ARRAY1[SI], 2DH          ;数字后面为减法，进入减法
JE SUBCAL
ADDCAL:
ADD AX, ARRAY1[SI+2]
ADD SI, 2
JMP CALCU
SUBCAL:
SUB AX, ARRAY1[SI+2]
ADD SI, 2
JMP CALCU

PRINT:
MOV DI,0
MOV BX,10
TEST AX,8000H                ;判断最高位
JZ P0                        ;正数
PUSH AX                      ;负数，输出'-'

MOV AX,0E0AH                 ;CRLF
INT 10H
MOV AX,0E0DH
INT 10H

MOV AX,0E2DH                 ;'-' ASCII
INT 10H
```

```
        POP AX
        NEG AX
        MOV DI,1

        P0:
        MOV DX,0
        XOR CX,CX
        Q0:
        XOR DX,DX
        DIV BX
        XOR DX,0E30H
        PUSH DX
        INC CX
        CMP AX,0
        JNZ Q0
        CMP DI,1
        JE Q1

        MOV AX,0E0AH                    ;CRLF
        INT 10H
        MOV AX,0E0DH
        INT 10H

        Q1:
        POP AX
        INT 10H
        LOOP Q1
        JMP ENDF

        ERROR:
        MOV DX, OFFSET ERRORINFO
        MOV AH, 9
        INT 21H

        ENDF:
        MOV AH, 4CH
        INT 21H
CODE ENDS
END START
```