



西安交通大学
XI'AN JIAOTONG UNIVERSITY

微机原理大作业

基于x86和华为服务器的 模拟计算器

小组成员：王靳朝 周帆 陈鑫旺

2023年5月17日

Contents Title



一、项目要求

二、基本运算式的实现

三、提高要求——计算器的实现

项目要求

基于x86和华为服务器的模拟计算器



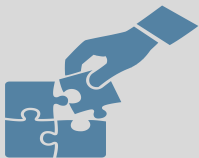
1.基本要求:

实现多项式运算，如： $Z=(X+Y)\times 8\div 2$ ，其中X,Y可自由设定值，验证结果，给出程序和调试结果图。
要求：用x86软件平台实现 $Z=(X+Y)\times 8\div 2$ 多项式功能。



2.提高要求:

在“基本要求”中内容的基础上实现计算器中任意两数+、-、 \times 、 \div 功能及其扩展，可采用子程序编制加减乘除，也可采用宏命令。



3.人机交互实现:

针对“提高要求”中内容，根据所选择的开发平台，完成下面的屏幕显示功能。
要求：使用DOS功能调用增加屏幕的输入显示，输出显示功能。

Contents Title



一、项目要求

二、基本运算式的实现

三、提高要求——计算器的实现

基于x86实现基本运算式

对于简单的多项式运算，程序顺序执行即可。

首先从键盘输入x和y作为运算变量，利用出栈、入栈指令，将x和y以低位起存入内存，一字节存入一位，最高十位。从低位到高位按字节进行 $x+y$ 和 $*8$ 的运算，并将运算结果存入中间变量；从高位到低位按字节进行 $/2$ 运算，并将计算结果从高位到低位依字节存入结果变量。最后将运算结果转换为对应的ASCII码，利用DOS功能调用显示在屏幕上。并设计简单的UI使人机交互更加清晰简便。

基本运算式的实现

部分关键代码

```
      ;输入变量Y,并逆序存入数据段
MOV AH,09H      ;提示用户输入变量Y
LEA DX,SVY
INT 21H
MOV BX,0      ;统计字节长
MOV CX,10      ;设置循环次数上限
LEA SI,VY      ;指针指向变量存储首地址
LIY1:  MOV AH,08H      ;逐字输入
INT 21H
CMP AL,0DH      ;检测,输入回车符表示输入完成
JZ JIENDY      ;完成,跳转
MOV DL,AL      ;显示
MOV AH,02H
INT 21H
SUB AL,30H      ;将ASCII码转换为十进制数,方便运算
PUSH AX      ;入栈
INC BX      ;计数
LOOP LIY1
JIENDY: CRLF
MOV CX,BX      ;设置循环次数为字节长
LIY2:  POP AX      ;出栈
MOV [SI],AL      ;未完成,则将字符存入
INC SI      ;指针步进
```

```
CADD      PROC NEAR      ;加法计算子程序
PUSH SI      ;保护现场
PUSH DI
PUSH BX
PUSH CX
PUSH AX
LEA SI,VX      ;指针SI指向变量X首地址
LEA DI,VY      ;指针DI指向变量Y首地址
LEA BX,VR      ;指针BX指向变量R首地址
MOV CX,10      ;设置循环次数
XOR AX,AX      ;AX清零
CLC
LADD:  MOV AL,[SI]      ;逐位相加
ADC AL,[DI]
AAA
MOV [BX],AL      ;存入结果
INC SI      ;指针步进
INC DI
INC BX
LOOP LADD
JNC JEND
MOV [BX],01H      ;进位计入
JEND:  POP AX      ;恢复现场
```

```
CDIV      PROC NEAR      ;除法计算子程序
PUSH DX      ;保护现场
PUSH DI
PUSH SI
PUSH AX
PUSH CX
MOV DL,02H      ;存入除数
LEA DI,VZ      ;指针DI指向变量Z首地址
LEA SI,VZ      ;指针SI指向变量R末地址
DEC SI
MOV AH,[SI]      ;AH装入
MOV AL,00H      ;AL清零
DEC SI
INC DI
MOV CX,11      ;设置循环次数(将第0、1位
        ;移出)
CLC
DIV:  MOV AL,[SI]      ;取出变量Z的字节
AAD
DIV DL      ;逐位相除
MOV [DI],AL      ;结果低四位存入AL
XOR AL,AL      ;AL清零
DEC SI      ;指针步退
INC DI      ;指针步进
LOOP LDIV
POP CX      ;恢复现场
POP AX
POP SI
POP DI
POP DX
RET
CDIV      ENDP
```



基本运算式的实现

运行结果示例

X=12,Y=9,得到Z=84

HCQC-A 微机接口 - USB接口设备实验(USB设备未发现) - [E:\My\仙交\课程\微机原理及应用\实验\myproject\calculator_1.ASM]

文件(F) 编辑(E) ASM文件编译(C) ASM文件调试(D) 设置 实验(S) 硬件检测(H) 工具软件(T) 窗口(W) 退出 帮助(H)

新建ASM 新建CPP 打开 保存 编译 构建 重构运行 结束调试 连续执行 单步执行 跳过 帮助 设置

```
0028 START:  MOV AX, DATA
0029          MOV DS, AX
0030          MOV AX, STACK1
0031          MOV SS, AX
0032          MOV AH, 09H
0033          LEA DX, SSE
0034          INT 21H
0035          MOV AH, 09H
0036          LEA DX, SPO
0037          INT 21H
0038          ;输入变量X,并逆序存入数据段
0039          MOV AH, 09H
0040          LEA DX, SVX
0041          INT 21H
0042          MOV BX, 0
0043          MOV CX, 10
0044          LEA SI, VX
0045 LIX1:    MOV AH, 08H
0046          INT 21H
0047          CMP AL, 0DH
0048          JZ JIENDX
0049          MOV DL, AL
```

C:\>d:
D:\>cd d:\exec
D:\EXEC>deltree /Y .
Deleting .\calculator.e...
D:\EXEC>..\all.com
loading file ... calculator_1.E
load & exec

Polynomial:Z=(X+Y)*8/2
X=
12
Y=
9
Z=
0000000000084
done!
D:\EXEC>
D:\EXEC>_

F12 enables mouse

调试关闭完成。
调试启动中,请稍候。。。
packing file E:\My\仙交\课程\微机原理及应用\实验\myproject\calculator_1.EXE(1260 bytes)...
调试启动完成。
连续执行命令中,请稍候。。。
dc

寄存器

名称	值
EAX	0x0000' 0FE6
EBX	0x0000' 8FEA
ECX	0x0014' 0000
EDX	0x0000' 2002
ESP	0x0000' FFFE
EBP	0x0000' 00EC
ESI	0x0000' 0100
EDI	0x0000' 0218
EIP	0x0000' 0000

反汇编

地址	反汇编指令
0FFF:0000	mov ax, 0xff6 ...
0FFF:0003	mov ds, ax ...
0FFF:0005	mov ax, 0xfe6 ...
0FFF:0008	mov ss, ax ...
0FFF:000A	mov ah, 0x9 ...
0FFF:000C	mov dx, 0x2d ...

栈察看

地址	值	显示
0FFF:0000	CD 20 C0 9F 00 9A F0 FE 1D F0 B1 02 38 0E 4A 01	
0FF7C:0010	38 0E 55 01 38 0E 38 0E 01 01 01 00 02 FF FF FF	
0FF7C:0020	FF FF FF FF FF FF FF FF FF FF FF FF 75 0F E6 FF	
0FF7C:0030	7C 0F 14 00 18 00 7C 0F FF FF FF FF 00 00 00 00	
0FF7C:0040	06 16 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
0FF7C:0050	CD 21 CB 00 00 00 00 00 00 00 00 00 00 20 20 20	
0FF7C:0060	20 20 20 20 20 20 20 20 00 00 00 00 00 00 20 20	

内存

地址	值
0FF7C:0000	CD 20 C0 9F 00 9A F0 FE 1D F0 B1 02 38 0E 4A 01
0FF7C:0010	38 0E 55 01 38 0E 38 0E 01 01 01 00 02 FF FF FF
0FF7C:0020	FF FF FF FF FF FF FF FF FF FF FF FF 75 0F E6 FF
0FF7C:0030	7C 0F 14 00 18 00 7C 0F FF FF FF FF 00 00 00 00
0FF7C:0040	06 16 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0FF7C:0050	CD 21 CB 00 00 00 00 00 00 00 00 00 00 20 20 20
0FF7C:0060	20 20 20 20 20 20 20 20 00 00 00 00 00 00 20 20

基于华为服务器实现基本运算式

利用C语言和ARM-V8混合语言编程的总体思路不变，可以使用C语言实现输出变量，并实现在屏幕上显示，可以直接使用C语言自带的运算符或者汇编语言实现多项式运算。由于本次着重训练软件接口技术，因此使用C语言和ARM-V8汇编语言混合编程实现多项式运算。

部分关键代码

```
int main()
{
    int x,y,z;
    printf("Please enter x=");
    scanf("%d",&x);
    printf("Please enter y=");
    scanf("%d",&y);
    printf("(x+y)*8/2=%d\n",calculate(x,y));
    return 0;
}

.global calculate
    mov w0,#0
calculate:
    add w0,w1,w0
    lsl w0,w0,3
    lsr w0,w0,1
    ret
```



基本运算式的实现

运行结果示例

```
Please enter x=12
```

```
Please enter y=34
```

```
(x+y)*8/2=184
```

```
xintong01@mcomp-interface:~$
```

```
Please enter x=3
```

```
Please enter y=9
```

```
(x+y)*8/2=48
```

```
xintong01@mcomp-interface:~$
```

```
Please enter x=100
```

```
Please enter y=9522
```

```
(x+y)*8/2=38488
```

```
xintong01@mcomp-interface:~$
```



Contents Title



一、项目要求

二、基本运算式的实现

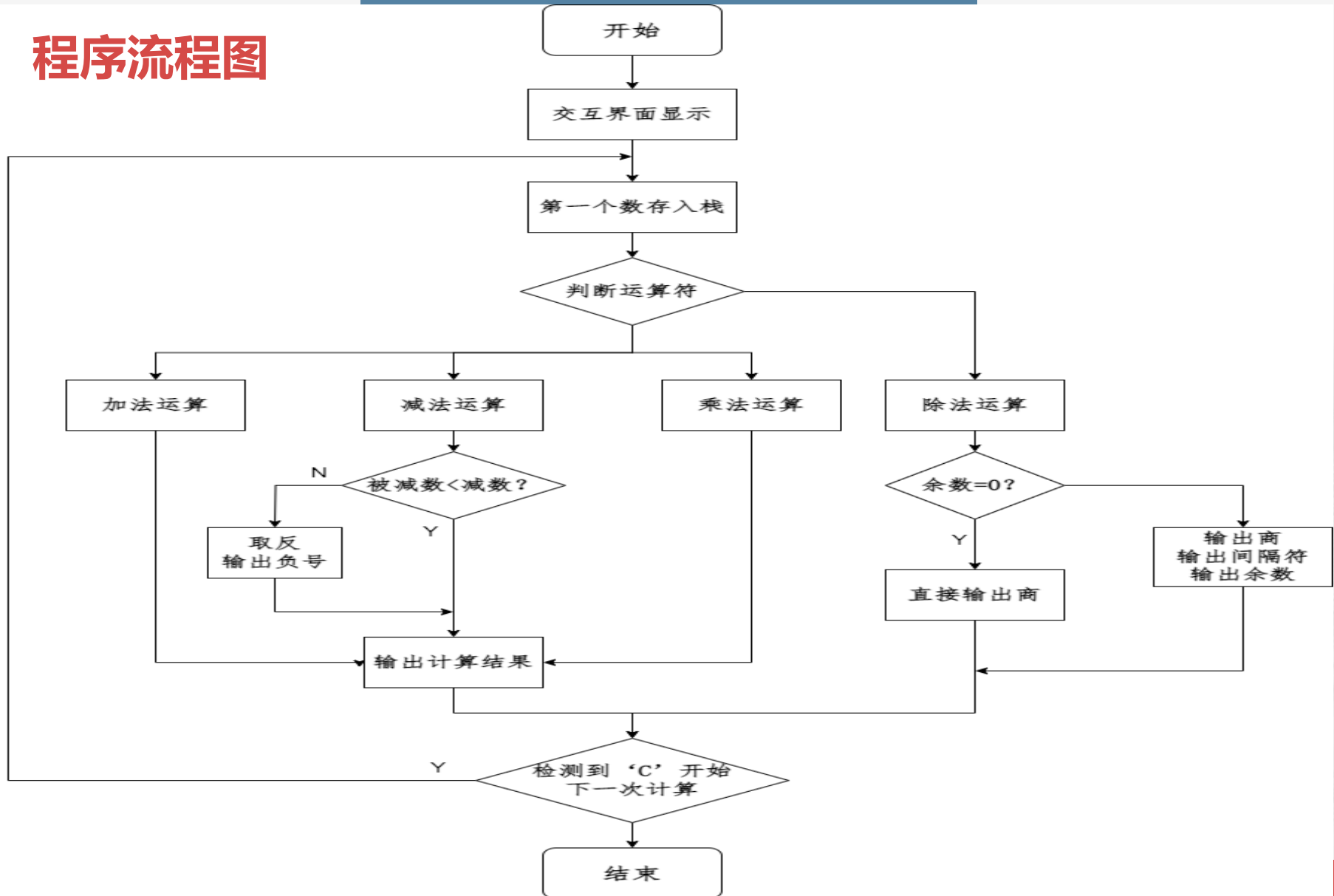
三、提高要求——计算器的实现

基于x86的计算器实现

开始运行程序之后首先显示交互界面，将第一个数存入堆栈，用户输入运算符进行判断，对于不同的运算符进行不同的操作，特别的对于减法运算需要判断结果是否为负数，对于除法运算需要判断是否存在余数，接着输入第二个数，存入寄存器，用户按下 '=' 运算并输出结果。最后检测用户是否按下 'C'，如果按下，则模拟计算器清0开始下一次运算，否则结束。

计算器实现

程序流程图



计算器实现

部分关键代码

```
REFRESH: XOR AX,AX  
CALL INPUT  
PUSH BX  
;判断运算符,跳转  
CMP AL,'+'  
JZ CADD  
CMP AL,'-'  
JZ CSUB  
CMP AL,'*'  
JZ CMUL  
CMP AL,'/'  
JZ CDIV
```

;AX清零
输入第一个
;存入栈

```
CADD: CALL INPUT ;输入第二个数  
POP AX ;第一个数取出  
ADD AX,BX ;相加  
JMP NEXT ;跳出  
CSUB: CALL INPUT ;输入第二个数  
POP AX ;第一个数取出  
CMP AX,BX ;判断被减数与减数的大小关系  
JL LESS ;小于,跳转  
SUB AX,BX ;大于,直接相减  
JMP NEXT  
LESS: SUB AX,BX ;相减为负则取反  
NEG AX  
PUSH AX ;保护AX  
MOV DL,'-' ;输出负号  
MOV AH,02H  
INT 21H  
POP AX ;送回AX  
JMP NEXT ;跳出  
CMUL: CALL INPUT ;输入第二个数  
POP AX ;第一个数取出  
MUL BX ;相乘  
JMP NEXT ;跳出
```

```
CDIV: CALL INPUT ;输入第二个数  
POP AX ;第一个数取出  
DIV BX ;相除  
CMP DX,0 ;检测余数  
JZ NEXT ;直接跳出  
PUSH DX ;保护DX  
CALL OUTPUT ;输出商  
MOV AH,09H ;显示间隔符  
LEA DX,SE  
INT 21H  
POP DX ;送回DX  
MOV AX,DX ;输出余数  
CALL OUTPUT  
JMP REFRESH ;跳出,刷新  
NEXT: CALL OUTPUT ;结果输出  
JMP REFRESH ;跳出,刷新  
MOV AH,4CH  
INT 21H
```



计算器实现

运行结果示例

输入27+9，输出36，键盘输入C之后进行下一次计算
特别的，整除运算27/9直接得到3；非整除27/8，得到商与余数3.....3

The screenshot displays the HQFC-A microcontroller interface, showing the assembly code for a calculator program and its execution results.

Assembly Code (Left Panel):

```
0029 START: MOV AX, DATA ;数据段
0030 MOV DS, AX
0031 MOV AX, STACK1 ;栈段
0032 MOV SS, AX
0033 MOV AH, 09H ;显示分割线
0034 LEA DX, SSE
0035 INT 21H
0036 MOV AH, 09H ;显示提示词
0037 LEA DX, STIP
0038 INT 21H
0039 REFRESH: XOR AX, AX ;AX清零
0040 CALL INPUT ;输入第一个
0041 PUSH BX ;存入栈
0042 ;判断运算符, 跳转
0043 CMP AL, '+'
0044 JZ CADD
0045 CMP AL, '-'
0046 JZ CSUB
0047 CMP AL, '*'
0048 JZ CMUL
0049 CMP AL, '/'
0050 JZ CDIV
```

Execution Results (Right Panel):

寄存器 (Registers):

名称	值
EAX	0x0000' 0FE6
EBX	0x0000' 8FEA
ECX	0x0000' 0000
EDX	0x0000' 2002
ESP	0x0000' FFFE
EBP	0x0000' 0054

反汇编 (Disassembly):

地址	反汇编指令
0FFC:0000	mov ax, 0xff6 ...
0FFC:0003	mov ds, ax ...
0FFC:0005	mov ax, 0xfe6 ...
0FFC:0008	mov ss, ax ...
0FFC:000A	mov ah, 0x9 ...

栈查看 (Stack View):

地址	值	显示
0FE6:FFFE	FF FF 00
0FE6:000E	00 00 00
0FE6:001E	00 00 00
0FE6:002E	00 00 00
0FE6:003E	00 00 00

内存 (Memory):

地址	值	显示
0F7C:0000	CD 20 C0
0F7C:0010	38 0E 55 ...	8.U.8.8....
0F7C:0020	FF FF FF
0F7C:0030	7C 0F 14

Command Window (Bottom):

```
Starting MS-DOS...
C:\>set path=c:\dos;d:\tasm\bin
C:\>d:
D:\>cd d:\exec
D:\EXEC>del tree /Y .
Deleting .\calculat.e...
D:\EXEC>..\all.com
loading file ... calculator_2.E
load & exec
*****
My Calculator:
27+9=36C
27-9=18C
9-27=-18C
27*9=243C
27/9=3C
27/8=3.....3C
-
F12 enables mouse
```

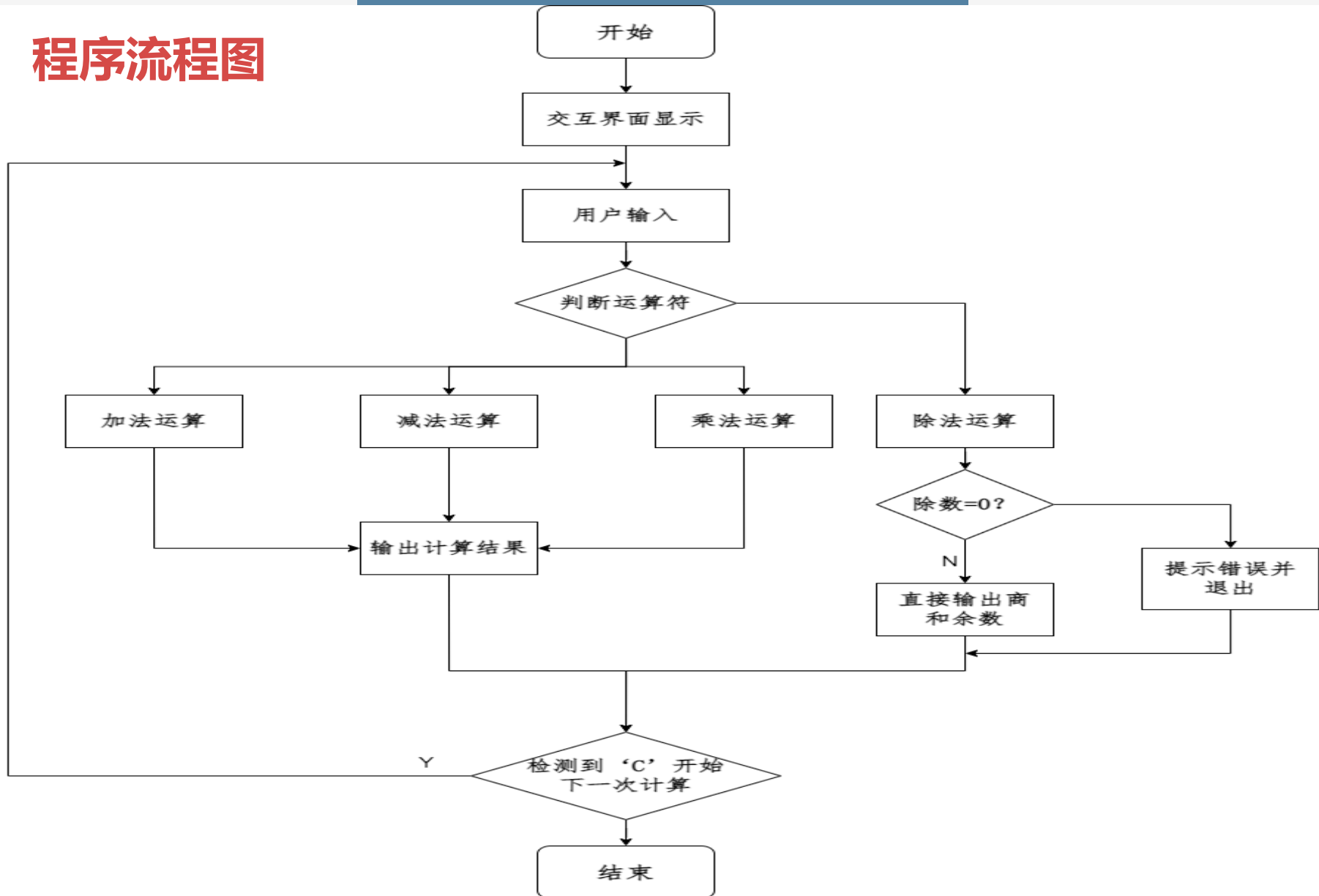
调试关闭完成。
调试启动中，请稍候。。。
packing file E:\My\仙交\课程\微机原理及应用\实验\myproject\calculator_2.EXE (1108 bytes)... done
调试启动完成。
连续执行命令中，请稍候。。。

基于华为服务器的计算器实现

华为服务器下使用C语言和ARM-V8混合语言编程。利用C语言可以构建基本的UI环境和输入输出函数，计算函数利用汇编语言实现。当程序开始运行之后首先显示UI界面，根据用户输入进行不同的操作，对错误的操作需要进行提示和退出。注意C语言和汇编语言的接口。每一次运算结束之后从键盘输入，检测是否为' C'，如果是，则继续下一次运算，从头开始运行，如果不是则退出计算器，因此使用do-while语句实现上述功能。

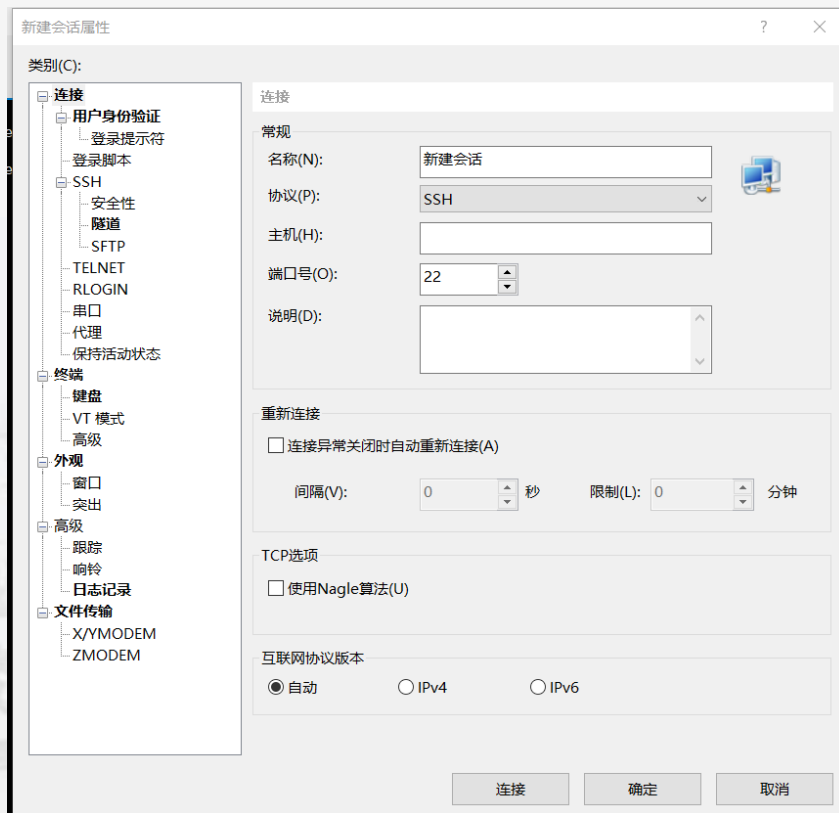
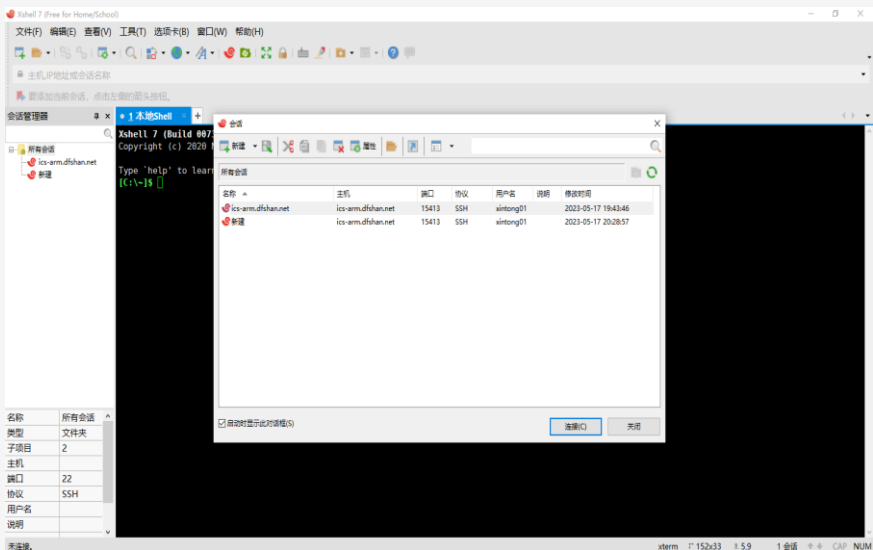
计算器实现

程序流程图



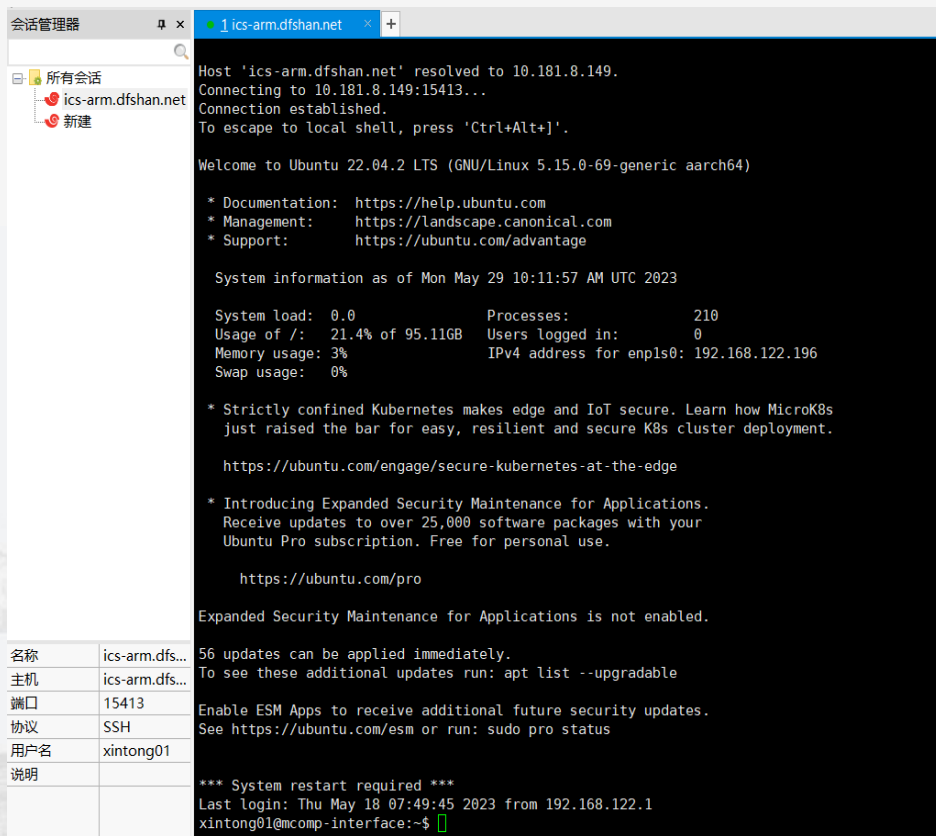
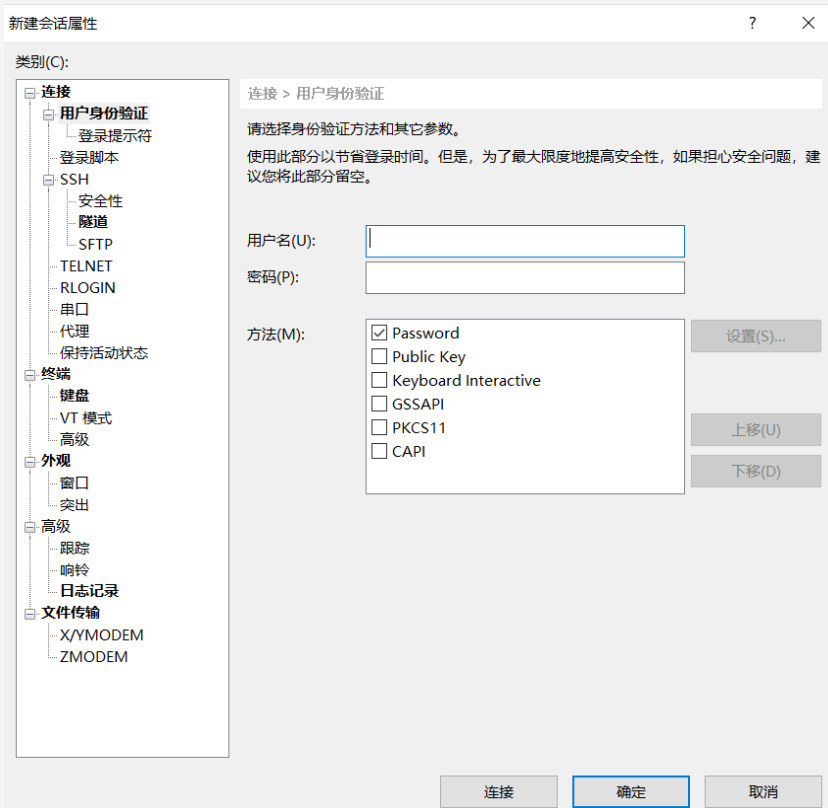
计算器实现

实验环境搭建



计算器实现

实验环境搭建



计算器实现

实验环境搭建

```
xintong01@mcomp-interface:~$  
xintong01@mcomp-interface:~$ nano calculator_2.c  
xintong01@mcomp-interface:~$ nano add.s  
xintong01@mcomp-interface:~$ nano sub.s  
xintong01@mcomp-interface:~$ nano mul.s  
xintong01@mcomp-interface:~$ nano div.s  
xintong01@mcomp-interface:~$
```

```
1 ics-arm.dfshan.net x +  
GNU nano 6.2  
global add  
    mov w0,#0  
add:  
    add w0,w0,w1  
    ret
```

```
1 ics-arm.dfshan.net x +  
GNU nano 6.2  
global sub  
    mov w0,#0  
sub:  
    sub w0,w0,w1  
    ret
```

```
1 ics-arm.dfshan.net x +  
GNU nano 6.2  
global mul  
    mov w0,#0  
mul:  
    mul w0,w0,w1  
    ret
```

```
1 ics-arm.dfshan.net x +  
GNU nano 6.2  
global div  
    mov w0,#0  
div:  
    sdiv w0,w0,w1  
    ret
```

```
GNU nano 6.2  
//-----  
//程序描述：可以实现任意两个数的加减乘除运算，输入输出  
//    均为int类型  
//作者：王新朝  
//时间：2023年5月16日  
//输入样例：UI界面显示之后，用户根据提示选择需要操作的类型  
//    例如，选择4号为除法运算，根据提示输入x=7, y=3  
//    运行结果之后显示商为2，余数为1。  
//    用户按下C之后模拟计算器清0，开始重新运算  
//调试工具：Xshell7 华为服务器  
//说明：无  
//-----  
#include<stdio.h>  
#include<stdlib.h>  
void menu()  
{  
    printf("|-----计算器-----|\n");  
    printf("|-----*1.Add*-----|\n");  
    printf("|-----*2.Sub*-----|\n");  
    printf("|-----*3.Mul*-----|\n");  
    printf("|-----*4.Div*-----|\n");  
    printf("|-----*0.Exit*-----|\n");  
}  
  
int add(int w0,int w1);  
int sub(int w0,int w1);  
int mul(int w0,int w1);  
extern div_t div(int w0,int w1);
```



计算器实现

部分关键代码

```
switch (a)
{
    case 1:
        printf("x+y=%d\n",add(x,y));
        break;
    case 2:
        printf("x-y=%d\n",sub(x,y));
        break;
    case 3:
        printf("x*y=%d\n",mul(x,y));
        break;
    case 4:
        if (y != 0)
        {
            printf("The quotient of x/y is %d\n",div(x,y));
            printf("The remainder of x/y is %d\n",x-(x/y)*y);
            break;
        }
        else
        {
            printf("Error!Please check the number of operands!\n");
        }
        default:
            printf("Invalid operation\n");
            break;
}
printf("Please Press 'C' to clear;press others to exit:\n");
scanf(" %c",&c);
}while(c=='C');
```

```
Add.s
.global add
        mov w0,#0

add:
        add w0,w0,w1
        ret

sub.s
.global sub
        mov w0,#0

sub:
        sub w0,w0,w1
        ret

mul.s
.global mul
        mov w0,#0

mul:
        mul w0,w0,w1
        ret

div.s
.global div
        mov w0,#0

div:
        sdiv w0,w0,w1
        ret
```



计算器实现

运行结果示例

首先测试退出功能：交互界面选择0号功能可以正常退出

```
xintong01@mcomp-interface:~$ ./calculator_2
|-----计算器-----|
|-----*1.Add*-----|
|-----*2.Sub*-----|
|-----*3.Mul*-----|
|-----*4.Div*-----|
|-----*0.Exit*-----|
Please select the operation type
0
Exit
```



计算器实现

运行结果示例

其次测试加法和减法功能以及能否正常开启下一次运算，选择加法和减法功能，检查发现结果均正确，同时运算结束之后键盘输入C键可以正常开启下一次运算。下一次运算需要重新选择操作类型并重新输入操作数。

```
xintong01@mcomp-interface:~$ ./calculator_2
|-----计算器-----|
|-----*1.Add*-----|
|-----*2.Sub*-----|
|-----*3.Mul*-----|
|-----*4.Div*-----|
|-----*0.Exit*-----|
Please select the operation type
1
Please enter the number of operands
x=-5
y=050
x+y=45
Please Press 'C' to clear,press others to exit:
C
Please select the operation type
2
Please enter the number of operands
x=90
y=1088
x-y=-998
Please Press 'C' to clear,press others to exit:
```



运行结果示例

检测运算符不合法的情况，可以正常提示并退出。

```
C
Please select the operation type
31
Please enter the number of operands
x=5
y=6
Invaild operation
```

乘法运算可以得出正确结果：

```
C
Please select the operation type
3
Please enter the number of operands
x=5
y=0
x*y=0
Please Press 'C' to clear,press others to exit:
C
```

运行结果示例

对于整除和非整除情况均可以得到正常结果并分别显示商和余数

:

```
C
Please select the operation type
4
Please enter the number of operands
x=64
y=8
The quotient of x/y is 8
The remainder of x/y is 0
Please Press 'C' to clear,press others to exit:
C
Please select the operation type
4
Please enter the number of operands
x=64
y=7
The quotient of x/y is 9
The remainder of x/y is 1
Please Press 'C' to clear,press others to exit:
```



运行结果示例

若除数为0，则可以正常提示错误信息。运算结束之后按下非' C' 按键可以正常退出。

```
C
Please select the operation type
4
Please enter the number of operands
x=18
y=0
Error!Please check the number of operands!
Invaild operation
Please Press 'C' to clear,press others to exit:
g
xintong01@mcomp-interface:~$
```





西安交通大学
XI'AN JIAOTONG UNIVERSITY

谢谢大家

