**汇编语言程序设计例子**

DATA SEGMENT

FUNCTION DB "THISIS A CALCULATOR.",0AH,0DH

DB "THIS PROGRAM IS USED TO PERFORM ",0AH,0DH

DB "PLUS, MINUS, MULITY, DIVIDE.",0AH,0DH,"$"

TISHI1 DB "PLEASE INPUT THE FIRST NUMBER:",0AH,0DH,"$"

TISHI2 DB 0ah,0dh,"PLEASE INPUT THE SECOND NUMBER:",0AH,0DH,"$"

TISHI3 DB 0ah,0dh,"PLEASE INPUT ONE OF +,-,\*,/:",0AH,0DH,"$"

NUMBER1 DB ?,?,?,?,?

NUMBER2 DB 5 DUP(?)

COUNTER1 DB 0

COUNTER2 DB 0

num1 dw 0

num2 dw 0

result dw ?

operator db ?

asc db 17 dup(0)

asccou dw 0

DATA ENDS

CODE SEGMENT

ASSUME CS:CODE,DS:DATA

S:MOV AX,DATA

MOV DS,AX

LEA DX,FUNCTION

MOV AH,9

INT 21H

LEA DX,TISHI1

MOV AH,9

INT 21H

mov cx,5

lea si,number1

call input

mov counter1,bl

mov cl,counter1

mov ch,0

lea si,number1

call convert

mov num1,ax

mov dl,0ah

mov ah,2

int 21h

mov dl,0dh

int 21h

inp:lea dx,tishi3

mov ah,9

int 21h

mov ah,1

int 21h

cmp al,'+'

je sec

cmp al,'-'

jz sec

cmp al,'\*'

jz sec

cmp al,'/'

jz sec

jmp inp

sec:mov operator,al

LEA DX,TISHI2

MOV AH,9

INT 21H

mov cx,5

lea si,number2

call input

mov counter2,bl

mov cl,counter2

mov ch,0

lea si,number2

call convert

mov num2,ax

mov al,operator

cmp al,'+'

je ladd

cmp al,'-'

jz lsub

cmp al,'\*'

jz lmul

cmp al,'/'

jz ldiv

ladd:mov ax,num1

add ax,num2

jmp lend

lsub:mov ax,num1

sub ax,num2

jmp lend

lmul:mov ax,num1

mul num2

jmp lend

ldiv:mov dx,0

mov ax,num1

div num2

lend:mov result,ax

mov ah,2

mov dl,0ah

int 21h

mov dl,0dh

int 21h

mov bx,result

lea di,operator

mov si,3

call chag

lea di,operator

mov cx,asccou

add di,cx

dec di

mov ah,2

disp:mov dl,[di]

int 21h

dec di

loop disp

MOV AH,4CH

INT 21H

input proc

mov bl,0

L:MOV AH,1

INT 21H

cmp al,0dh

jz L1

mov [si],al

inc si

inc bl

loop L

L1:ret

input endp

convert proc

mov ax,0

mov bx,10

l3:mul bx

sub byte ptr [si],30h

add al,[si]

adc ah,0

inc si

loop l3

ret

convert endp

bin proc

mov si,16

mov cl,1

out1:rol bx,cl

mov dl,bl

and dl,01h

add dl,30h

mov [di],dl

inc di

dec si

cmp si,0

jnz out1

mov byte ptr [di],'$'

ret

bin endp

hex proc

mov si,4

mov cl,4

hex\_out1:rol bx,cl

mov dl,bl

and dl,0fh

cmp dl,9

jle deci

add dl,7

deci:add dl,30h

mov [di],dl

inc di

dec si

cmp si,0

jnz hex\_out1

mov byte ptr [di],'$'

ret

hex endp

qua proc

mov si,5

add di,6

mov byte ptr [di],'$'

mov cl,3

qua\_out1:mov dl,bl

and dl,07

add dl,30h

dec di

mov [di],dl

shr bx,cl

dec si

cmp si,0

jnz qua\_out1

dec di

and bl,7

add bl,30h

mov [di],bl

ret

qua endp

decx proc

add di,5

mov byte ptr [di],'$'

mov si,5

mov ax,bx

dec\_out1:mov dx,0

mov cx,10

div cx

add dl,30h

dec di

mov [di],dl

dec si

cmp si,0

jnz dec\_out1

ret

decx endp

chag proc

mov cx,0

mov ax,bx

dec\_out2:mov dx,0

div si

cmp dl,9

jle deci1

add dl,7

deci1:add dl,30h

mov [di],dl

inc di

inc cx

cmp ax,0

jnz dec\_out2

mov asccou,cx

ret

chag endp

CODE ENDS

END S