

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ
по лабораторной работе №3
по дисциплине «Организация ЭВМ и систем»
Тема: Представление и обработка целых чисел. Организация
ветвящихся процессов

Студент гр. 0384

Кусмарцев А.И.

Преподаватель

Ефремов М. А.

Санкт-Петербург

2021

Цель работы.

Научиться применять метки и команду `str` для ветвления команд в программе.

Задание.

Разработать на языке Ассемблера программу, которая по заданным целочисленным значениям параметров a , b , i , k вычисляет:

а) значения функций $i1 = f1(a,b,i)$ и $i2 = f2(a,b,i)$

б) значения результирующей функции $res = f3(i1,i2,k)$, где вид функций $f1$ и $f2$ определяется из табл. 2, а функции $f3$ - из табл.3 по цифрам шифра индивидуального задания ($n1,n2,n3$), приведенным в табл.4. Значения a , b , i , k являются исходными данными, которые должны выбираться студентом самостоятельно и задаваться в процессе исполнения программы в режиме отладки. При этом следует рассмотреть всевозможные комбинации параметров a , b и k , позволяющие проверить различные маршруты выполнения программы, а также различные знаки параметров a и b .

Выполнение работы.

Вариант 13

$/ - (4*i+3)$, при $a>b$

$f2 = <$

$\backslash 6*i -10$, при $a\leq b$

$/ - (6*i+8)$, при $a>b$

$f8 = <$

$\backslash 9 -3*(i-1)$, при $a\leq b$

$/ |i1 + i2|$, при $k=0$

$f3 = <$

$\backslash \min(i1,i2)$, при $k\neq 0$

В работе использовались команды для условный переходов такие как: jle и je, также использовалась команда stp, которая изменяла состояния флагов SF, ZF, OF, на основе которых выполнялись команды условного перехода. Проверим работу программы в отладчике

Значения a, b, i, k	Результат, который записывается в i1	Результат, который записывается в i2	Результат, который записывается в res	Комментарий
a=1 b=1 i =1 k =1	FFFC = -4	0009 = 9	FFFC = -4	Верно
a = 0 b = 1 i = 2 k = 0	0002 = 2	0006 = 6	0008=8	Верно
A = 2 b = 1 i = 3 k = 1	FFF1 = -15	FFE6 = -26	FFE6 = -26	Верно
a = 2 b = 1 i = 4 k = 0	FFED = - 19	FFE0 = -32	0033=51	Верно

Разработанный программный код см. в приложении А.

Выводы.

В ходе выполнения работы были изучена работа ветвления команд в языке ассемблера

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lab3.asm

```
Astack SEGMENT STACK
        DW 12 DUP(?)
Astack ENDS
```

```
DATA SEGMENT
a      DW 0
b      DW 0
i      DW 0
k      DW 0
i1     DW 0
i2     DW 0
res    DW 0
```

```
DATA ENDS
```

```
CODE SEGMENT
        ASSUME CS:CODE, DS:DATA, SS:Astack
Main PROC FAR
        push ds
        sub ax, ax
        mov ax, DATA
        mov ds, ax
            mov a, 2
            mov b, 1
            mov i, 4
            mov k, 0
        mov cx, i
        shl cx, 1
        shl cx, 1
        mov ax, cx
        mov bx, b
        cmp a, bx
        jle f1
            ;a>b
            ;i1
            neg ax
            mov cx, 3
            neg cx
            add cx, ax
            mov i1, cx
            ;i2
            mov cx, i
            shl cx, 1
            neg cx
            add ax, cx
            mov i2, ax
            mov cx, 8
            neg cx
            add cx, ax
            mov i2, cx
```

```

        jmp f1end

f1:      ;a<=b
        ;i1
        mov cx, i
        shl cx, 1
        add ax, cx
        mov cx, 10
        neg cx
        add cx, ax
        mov i1, cx
        ;i2
        mov cx, i
        shl cx, 1
        add cx, i
        neg cx
        add cx, 12
        mov i2, cx

f1end:
        mov bx, i2
        cmp k, 0
        je f2
        ; k != 0
        cmp i1, bx
        jle f3
        ; i1 > i2
        mov ax, i2
        jmp AllEnd

        f3:  ;i1 <= i2
        mov ax, i1
        jmp AllEnd

f2:      ;k==0
        mov ax, i1
        add ax, i2
        cmp ax, 0
        jle f4
        ; i1+i2 > 0
        jmp AllEnd
        f4: ;i1+i2 <= 0
        neg ax

AllEnd:
        mov res, ax
        ret

Main ENDP
CODE ENDS
END Main

```

Название файла: lab3.lst

#Microsoft (R) Macro Assembler Version 5.10
11/25/21 01:18:3

Page

1-1

```
0000          Astack SEGMENT STACK
0000 000C[          DW 12 DUP(?)
      ????
      ]

0018          Astack ENDS

0000          DATA SEGMENT
0000 0000          a      DW      0
0002 0000          b      DW      0
0004 0000          i      DW      0
0006 0000          k      DW      0
0008 0000          i1     DW      0
000A 0000          i2     DW      0
000C 0000          res     DW      0

000E          DATA ENDS

0000          CODE SEGMENT
      ASSUME CS:CODE, DS:DATA, SS:Astack
0000 Main PROC FAR
0000 1E          push ds
0001 2B C0          sub ax, ax
0003 B8 ---- R      mov ax, DATA
0006 8E D8          mov ds, ax
0008 C7 06 0000 R 0002      mov a, 2
000E C7 06 0002 R 0001      mov b, 1
0014 C7 06 0004 R 0004      mov i, 4
001A C7 06 0006 R 0000      mov k, 0
0020 8B 0E 0004 R      mov cx, i
0024 D1 E1          shl cx, 1
0026 D1 E1          shl cx, 1
0028 8B C1          mov ax, cx
002A 8B 1E 0002 R      mov bx, b
002E 39 1E 0000 R      cmp a, bx
0032 7E 28          jle f1
      ;a>b
      ;i1

0034 F7 D8          neg ax
0036 B9 0003          mov cx, 3
0039 F7 D9          neg cx
003B 03 C8          add cx, ax
003D 89 0E 0008 R      mov i1, cx
      ;i2

0041 8B 0E 0004 R      mov cx, i
0045 D1 E1          shl cx, 1
```

```

0047 F7 D9          neg cx
0049 03 C1          add ax, cx
004B A3 000A R      mov i2, ax
004E B9 0008        mov cx, 8
0051 F7 D9          neg cx
0053 03 C8          add cx, ax
0055 89 0E 000A R   mov i2, cx

```

#Microsoft (R) Macro Assembler Version 5.10
01:18:3

11/25/21

Page

1-2

```

0059 EB 27 90          jmp f1end

005C                  f1:  ;a<=b
                        ;i1
005C 8B 0E 0004 R      mov cx, i
0060 D1 E1            shl cx, 1
0062 03 C1            add ax, cx
0064 B9 000A          mov cx, 10
0067 F7 D9            neg cx
0069 03 C8            add cx, ax
006B 89 0E 0008 R     mov i1, cx
                        ;i2
006F 8B 0E 0004 R     mov cx, i
0073 D1 E1            shl cx, 1
0075 03 0E 0004 R     add cx, i
0079 F7 D9            neg cx
007B 83 C1 0C         add cx, 12
007E 89 0E 000A R     mov i2, cx

0082                  f1end:
0082 8B 1E 000A R      mov bx, i2
0086 83 3E 0006 R 00   cmp k, 0
008B 74 12            je f2
                        ; k != 0
008D 39 1E 0008 R      cmp i1, bx
0091 7E 06            jle f3
                        ;i1 > i2
0093 A1 000A R          mov ax, i2
0096 EB 18 90          jmp AllEnd

0099                  f3:  ;i1 <= i2
0099 A1 0008 R          mov ax, i1
009C EB 12 90          jmp AllEnd

009F                  f2: ;k==0
009F A1 0008 R          mov ax, i1
00A2 03 06 000A R     add ax, i2
00A6 3D 0000          cmp ax, 0
00A9 7E 03            jle f4
                        ; i1+i2 > 0

```



```

00AB EB 03 90                                jmp AllEnd
00AE                                f4: ;i1+i2 <= 0

00AE F7 D8                                neg ax

```

```

00B0                                AllEnd:
00B0 A3 000C R                        mov res, ax
00B3 CB                            ret
00B4                                Main ENDP
00B4                                CODE ENDS
#Microsoft (R) Macro Assembler Version 5.10
01:18:3

```

11/25/21

Page

1-3

END Main

```

#Microsoft (R) Macro Assembler Version 5.10
01:18:3

```

11/25/21

Symbols-1

Segments and Groups:

	N a m e	Length	Align	Combine	Class
ASTACK	0018	PARA	STACK	
CODE	00B4	PARA	NONE	
DATA	000E	PARA	NONE	

Symbols:

	N a m e	Type	Value	Attr	
A	L WORD	0000	DATA	
ALLEND	L NEAR	00B0	CODE	
B	L WORD	0002	DATA	
F1	L NEAR	005C	CODE	
F1END	L NEAR	0082	CODE	
F2	L NEAR	009F	CODE	
F3	L NEAR	0099	CODE	
F4	L NEAR	00AE	CODE	
I	L WORD	0004	DATA	
I1	L WORD	0008	DATA	
I2	L WORD	000A	DATA	
K	L WORD	0006	DATA	
MAIN	F PROC	0000	CODE	Length =
	00B4				

RES	L WORD	000C DATA
@CPU	TEXT	0101h
@FILENAME	TEXT	lab3
@VERSION	TEXT	510

102 Source Lines
102 Total Lines
22 Symbols

48056 + 461251 Bytes symbol space free

0 Warning Errors
0 Severe Errors