

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

Студентка гр. 1381

Тулегенова А.О.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

Цель работы.

Изучить представление и обработку целых чисел, научиться организовывать ветвящиеся процессы.

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 .

Вариант 25:

$$\begin{aligned} f5 &= \begin{cases} 20 - 4i, & \text{при } a > b \\ -(6i - 6), & \text{при } a \leq b \end{cases} \\ f8 &= \begin{cases} -(6i + 8), & \text{при } a > b \\ 9 - 3(i - 1), & \text{при } a \leq b \end{cases} \\ f6 &= \begin{cases} |i1 - i2|, & \text{при } k < 0 \\ \max(7, |i2|), & \text{при } k \geq 0 \end{cases} \end{aligned}$$

Ход выполнения работы

Были заданы сегменты данных под ДОС, определена модель памяти small, отведена память под размер стека, проинициализированы переменные a, b, i, k . Далее было загружено в ds начало сегмента данных, происходит сравнение переменных a и b с помощью команды `cmp`. Если $a > b$, то с помощью команды `jg` выполняется переход к метке `func2`, иначе выполняется метка `func1`.

Далее происходит вычисление значений $i1$ и $i2$ с помощью команд арифметического сдвига, сложения и вычитания, данные значения записываются в регистры ax и sx . Далее переход к метке `func3`, значение переменной k сравнивается с нулем и если он больше или равен нулю, то происходит переход к метке `max`, в которой берется по модулю значение $i2$ и оно сравнивается с 7, большее значение записывается в регистр sx . Если значение переменной k меньше нуля, то происходит переход к метке `abs`, в которой в регистр sx записывается модуль разности между значениями $i1$ и $i2$. Далее переходим к метке `result`, в переменную `res` записывается значение регистра sx . После этого программа завершается.

Таблица 1. Тестирование работы программы.

Значение констант	Ожидаемый результат	Полученный результат
$a = 1$ $b = 2$ $i = 3$ $k = 4$	$i1 = -12$ (FFF4) $i2 = 3$ (0003) $f3 = 7$ (0007)	$i1 = -12$ (FFF4) $i2 = 3$ (0003) $f3 = 7$ (0007)
$a = 3$ $b = 1$ $i = -1$ $k = 0$	$i1 = 24$ (0018) $i2 = -2$ (FFF2) $f3 = 7$ (0007)	$i1 = 24$ (0018) $i2 = -2$ (FFFE) $f3 = 7$ (0007)
$a = 2$ $b = 2$ $i = 0$ $k = -1$	$i1 = 6$ (0006) $i2 = 12$ (000C) $f3 = 6$ (0006)	$i1 = 6$ (0006) $i2 = 12$ (000C) $f3 = 6$ (0006)

Вывод

При выполнении лабораторной работы было изучено представление и обработка целых чисел, организация ветвящихся процессов.

ПРИЛОЖЕНИЕ А

КОД ПРОГРАММЫ

Название файла: lab3.asm

```
dosseg
.model small
.stack 100h
.data
    a dw 3
    b dw 1
    i dw -1
    k dw 0
    res dw ?

.code
mov ax, @data
mov ds, ax
mov ax, a
cmp ax, b
jg func2

func1:    ; a<=b
    mov ax, i    ; i
    shl ax, 1    ; 2i
    add ax, i    ; 3i
    shl ax, 1    ; 6i
    sub ax, 6    ; 6i-6
    neg ax       ; -(6i-6)

    mov cx, i    ; i
    shl cx, 1    ; 2i
    add cx, i    ; 3i
    neg cx       ; -3i
    add cx, 12   ; 9-3(i-1)
    jmp func3

func2:    ; a>b
    mov ax, i    ; i
    add ax, ax   ; 2i
    add ax, ax   ; 4i
    neg ax       ; -4i
```

```

        add ax, 20      ; 20-4i

        mov cx, i       ; i
        shl cx, 1       ; 2i
        add cx, i       ; 3i
        shl cx, 1       ; 6i
        add cx, 8       ; 6i+8
        neg cx          ; -(6i+8)

func3:
        cmp k, 0
        jge max
        sub cx, ax      ; i2-i1

abs:     ; k<0
        neg cx
        js abs          ; |i1-i2|
        jmp result

max:     ; k>=0
        neg cx          ; -i2
        js max          ; |i2|
        cmp cx, 7
        jge result
        mov cx, 7
        jmp result

result:
        mov res, cx
        mov ah, 4ch
        int 21h
        end

```

ПРИЛОЖЕНИЕ Б

ФАЙЛЫ ДИАГНОСТИЧЕСКИХ СООБЩЕНИЙ

Название файла: lab3.lst

Microsoft (R) Macro Assembler Version 5.10
10/31/22 00:18:3

Page 1-1

```
dosseg
.model small
.stack 100h
.data

0000 0003          a dw 3
0002 0001          b dw 1
0004 FFFF          i dw -1
0006 0000          k dw 0
0008 0000          res dw ?

.code

0000 B8 ---- R      mov ax, @data
0003 8E D8          mov ds, ax
0005 A1 0000 R      mov ax, a
0008 3B 06 0002 R    cmp ax, b
000C 7F 22          jg func2

000E              func1:      ; a<=b
000E A1 0004 R      mov ax, i      ; i
0011 D1 E0          shl ax, 1      ; 2i
0013 03 06 0004 R    add ax, i      ; 3i
0017 D1 E0          shl ax, 1      ; 6i
0019 2D 0006          sub ax, 6      ; 6i-6
001C F7 D8          neg ax          ; -(6i-6)

001E 8B 0E 0004 R    mov cx, i      ; i
0022 D1 E1          shl cx, 1      ; 2i
0024 03 0E 0004 R    add cx, i      ; 3i
0028 F7 D9          neg cx          ; -3i
002A 83 C1 0C          add cx, 12      ; 9-3(i-1)
002D EB 1E 90          jmp func3
```

```

0030                                func2:    ; a>b
0030  A1 0004 R                      mov ax, i    ; i
0033  03 C0                          add ax, ax   ; 2i
0035  03 C0                          add ax, ax   ; 4i
0037  F7 D8                          neg ax     ; -4i
0039  05 0014                        add ax, 20    ; 20-4i

003C  8B 0E 0004 R                  mov cx, i    ; i
0040  D1 E1                          shl cx, 1    ; 2i
0042  03 0E 0004 R                  add cx, i    ; 3i
0046  D1 E1                          shl cx, 1    ; 6i
0048  83 C1 08                      add cx, 8     ; 6i+8
004B  F7 D9                          neg cx     ; -(6i+8)

```

```

004D                                func3:
004D  83 3E 0006 R 00                cmp k, 0
0052  7D 09                          jge max
0054  2B C8                          sub cx, ax   ; i2-i1

```

```

0056                                abs:    ; k<0
0056  F7 D9                          neg cx
0058  78 FC                          js abs     ; |i1-i2|

```

Microsoft (R) Macro Assembler Version 5.10

10/31/22 00:18:3

Page 1-2

```

005A  EB 10 90                      jmp result

```

```

005D                                max:    ; k>=0
005D  F7 D9                          neg cx     ; -i2
005F  78 FC                          js max     ; |i2|
0061  83 F9 07                      cmp cx, 7
0064  7D 06                          jge result
0066  B9 0007                        mov cx, 7
0069  EB 01 90                      jmp result

```

```

006C                                result:
006C  89 0E 0008 R                  mov res, cx

```

```

0070  B4 4C          mov ah, 4ch
0072  CD 21          int 21h

```

end

```

Microsoft      (R)      Macro      Assembler      Version      5.10
10/31/22 00:18:3

```

Symbols-1

Segments and Groups:

Class	N a m e	Length	Align	Combine
DGROUP	GROUP		
_DATA	000A WORD PUBLIC		'DATA'
STACK	0100 PARA STACK		'STACK'
_TEXT	0074 WORD PUBLIC		'CODE'

Symbols:

	N a m e	Type	Value	Attr
A	L WORD	0000	_DATA
ABS	L NEAR	0056	_TEXT
B	L WORD	0002	_DATA
FUNC1	L NEAR	000E	_TEXT
FUNC2	L NEAR	0030	_TEXT
FUNC3	L NEAR	004D	_TEXT
I	L WORD	0004	_DATA
K	L WORD	0006	_DATA
MAX	L NEAR	005D	_TEXT
RES	L WORD	0008	_DATA
RESULT	L NEAR	006C	_TEXT

@CODE	TEXT	_TEXT
@CODESIZE	TEXT	0
@CPU	TEXT	0101h
@DATASIZE	TEXT	0
@FILENAME	TEXT	lab3
@VERSION	TEXT	510

69 Source Lines
69 Total Lines
28 Symbols

47994 + 459266 Bytes symbol space free

0 Warning Errors
0 Severe Errors