

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

ОТЧЕТ
по лабораторной работе №3
по дисциплине «Организация ЭВМ и систем»
Тема: Представление и обработка целых чисел. Организация
ветвящихся процессов
Вариант № 26

Студент гр.1381

—

Хомутильников Н.А.

Преподаватель

—

Ефремов М. А.

Санкт-Петербург

2022

Цель работы.

Изучить представление и обработку целых чисел на языке Ассемблер.
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 №26

$f6 = \begin{cases} / 2*(i+1)-4, & \text{при } a > b \\ \backslash 5-3*(i+1), & \text{при } a \leq b \end{cases}$	$f8 = \begin{cases} / -(6*i+8), & \text{при } a > b \\ \backslash 9-3*(i+1), & \text{при } a \leq b \end{cases}$	$f1 = \begin{cases} / \min(i1, i2), & \text{при } k=0 \\ \backslash \max(i1, i2) & \text{при } k \neq 0 \end{cases}$
---	--	--

Выполнение работы.

1. Объявлена упрощенная модель сегментации типа SMALL. Под стек отведено 256 байт. Исходный код программы см. в приложении А.
2. В сегменте данных (.data) были объявлены однобайтные переменные a , b , i , k , $i1$, $i2$, res .
3. В сегменте кода (.code) адрес сегмента данных помещается в регистр ds , после чего начинается работа с функциями. Были использованы регистры al , bl , cl , dl , ax . Для выполнения задания при реализации функций использовались следующие команды:

- 1) JMP (JUMP) – команда дальнего или ближнего перехода к метке

2) JL (Jump if less) – команда, выполняющая короткий переход к метке, если при сравнении командой `cmp` первый операнд меньше второго.

3) JLE (Jump if less or equal) – команда, выполняющая короткий переход, если при сравнении командой `cmp` первый операнд меньше или равен второму.

4) JNE (Jump if not equal) - команда, выполняющая короткий переход, если при сравнении командой `cmp` первый операнд не равен второму.

5) JE (Jump if equal) - команда, выполняющая короткий переход, если при сравнении командой `cmp` первый операнд равен второму.

6) NEG – меняет знак числа

Тестирование.

Чтобы проверить корректность работы программы, было проведено три

1. Результаты работы программы при $a=5$; $b=3$; $i=10$; $k=0$ представлены в табл.1.

i1	i2	res	Итог
12 (18)	BC (-68)	BC (-68)	Верно

Таблица 1 – Результаты первого теста

2. Результаты работы программы при $a=3$; $b=5$; $i=10$; $k=0$ представлены в табл.2.

i1	i2	res	Итог
E4 (-28)	EE (-18)	E4 (-28)	Верно

Таблица 2 – Результаты второго теста

3. Результаты работы программы при $a=15$; $b=15$; $i=10$; $k=1$ представлены в табл.3.

i1	i2	res	Итог
E4 (-28)	EE (-18)	EE (-18)	Верно

Таблица 3 – Результаты третьего теста

Выводы.

В ходе выполнения лабораторной работы было изучено представление и обработка целых чисел, и организация ветвящихся процессов. Для выполнения задания была написана программа, которая вычисляет значения функций согласно заданным условиям.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММ

Название файла: *lab3.asm*

```
DOSSEG
.MODEL SMALL
.STACK 100H

.DATA

    a db 5
    b db 3
    i db 10
    k db 0

    i1 db ?
    i2 db ?
    res db ?

.CODE

    mov ax, @data
    mov ds, ax
    mov al, a
    cmp al, b
    jle another

;f6_1 (2*(i+1)-4)
    mov bl, i
    add bl, 1
    mov ax, 2
    mul bl
    sub al, 4
    mov i1, al

;f8_1 (-(6*i+8))
    mov bl, i
    mov ax, 6
    mul bl
    add al, 8
    neg al
    mov i2, al

    cmp k, 0
    jmp f1_1

    another:
;f6_2 (5-3*(i+1))
    mov bl, i
```

```

    add bl, 1
    mov ax, 3
    mul bl
    neg al
    add al, 5
    mov i1, al
;f8_2 (9-3*(i-1))
    mov bl, i
    sub bl, 1
    mov ax, 3
    mul bl
    neg al
    add al, 9
    mov i2, al

    jmp f1_1

f1_1:
;min(i1, i2)
    mov al, i1
    mov bl, i2
    mov cl, k
    cmp k, 0
    jne f1_2
    cmp al, bl
    jl min
    je equal
    mov dl, i2
    jmp ending

f1_2:
;max(i1, i2)

    mov al, i1
    mov bl, i2
    cmp bl, al
    jg max
    je equal
    mov dl, i1
    jmp ending

min:
    mov dl, i1
    mov res, dl
    jmp ending

equal:
    mov dl, i1
    mov res, dl
    jmp ending

max:
    mov dl, i2

```

```

mov res, dl
jmp ending

```

```

ending:
mov ah, 4ch
int 21h

```

END

Название файла: *lab3.lst*

Microsoft (R) Macro Assembler Version 5.10
02:43:07

12/4/22

Page 1-

1

```

DOSSEG
.MODEL SMALL
.STACK 100H

```

.DATA

```

0000 05          a db 5
0001 03          b db 3
0002 0A          i db 10
0003 00          k db 0

0004 00          i1 db ?
0005 00          i2 db ?
0006 00          res db ?

```

.CODE

```

0000 B8 ---- R      mov ax, @data
0003 8E D8          mov ds, ax
0005 A0 0000 R      mov al, a
0008 3A 06 0001 R   cmp al, b
000C 7E 29          jle another

```

```

                                ;f6_1 (2*(i+1)-4)
000E 8A 1E 0002 R   mov bl, i
0012 80 C3 01          add bl, 1
0015 B8 0002          mov ax, 2
0018 F6 E3          mul bl
001A 2C 04          sub al, 4
001C A2 0004 R      mov i1, al

```

```

                                ;f8_1 (-(6*i+8))
001F 8A 1E 0002 R   mov bl, i

```

```

0023 B8 0006          mov ax, 6
0026 F6 E3          mul bl
0028 04 08          add al, 8
002A F6 D8          neg al
002C A2 0005 R      mov i2, al

002F 80 3E 0003 R 00      cmp k, 0
0034 EB 2A 90          jmp f1_1

0037                      another:
                      ;f6_2 (5-3*(i+1))
0037 8A 1E 0002 R      mov bl, i
003B 80 C3 01          add bl, 1
003E B8 0003          mov ax, 3
0041 F6 E3          mul bl
0043 F6 D8          neg al
0045 04 05          add al, 5
0047 A2 0004 R      mov i1, al
                      ;f8_2 (9-3*(i-1))
004A 8A 1E 0002 R      mov bl, i

```


2

```
004E 80 EB 01          sub bl, 1
0051 B8 0003          mov ax, 3
0054 F6 E3          mul bl
0056 F6 D8          neg al
0058 04 09          add al, 9
005A A2 0005 R       mov i2, al

005D EB 01 90          jmp f1_1

0060                  f1_1:
                      ;min(i1, i2)
0060 A0 0004 R       mov al, i1
0063 8A 1E 0005 R   mov bl, i2
0067 8A 0E 0003 R   mov cl, k
006B 80 3E 0003 R 00 cmp k, 0
0070 75 0D          jne f1_2
0072 3A C3          cmp al, bl
0074 7C 1D          jl min
0076 74 26          je equal
0078 8A 16 0005 R   mov dl, i2
007C EB 36 90          jmp ending

007F                  f1_2:
                      ;max(i1, i2)
007F A0 0004 R       mov al, i1
0082 8A 1E 0005 R   mov bl, i2
0086 3A D8          cmp bl, al
0088 7F 1F          jg max
008A 74 12          je equal
008C 8A 16 0004 R   mov dl, i1
0090 EB 22 90          jmp ending

0093                  min:
0093 8A 16 0004 R   mov dl, i1
0097 88 16 0006 R   mov res, dl
009B EB 17 90          jmp ending

009E                  equal:
009E 8A 16 0004 R   mov dl, i1
00A2 88 16 0006 R   mov res, dl
00A6 EB 0C 90          jmp ending

00A9                  max:
00A9 8A 16 0005 R   mov dl, i2
00AD 88 16 0006 R   mov res, dl
00B1 EB 01 90          jmp ending
```

<i>00B4</i>		<i>ending:</i>
<i>00B4</i>	<i>B4 4C</i>	<i>mov ah, 4ch</i>
<i>00B6</i>	<i>CD 21</i>	<i>int 21h</i>

Microsoft (R) Macro Assembler Version 5.10
02:43:07

12/4/22

Page 1-

3

END

Segments and Groups:

	N a m e	Length	Align	Combine Class
DGROUP	GROUP		
_DATA	0007 WORD PUBLIC		'DATA'
_STACK	0100 PARA STACK		'STACK'
_TEXT	00B8 WORD PUBLIC		'CODE'

Symbols:

	N a m e	Type	Value	Attr
A	L BYTE	0000	_DATA
ANOTHER	L NEAR	0037	_TEXT
B	L BYTE	0001	_DATA
ENDING	L NEAR	00B4	_TEXT
EQUAL	L NEAR	009E	_TEXT
F1_1	L NEAR	0060	_TEXT
F1_2	L NEAR	007F	_TEXT
I	L BYTE	0002	_DATA
I1	L BYTE	0004	_DATA
I2	L BYTE	0005	_DATA
K	L BYTE	0003	_DATA
MAX	L NEAR	00A9	_TEXT
MIN	L NEAR	0093	_TEXT
RES	L BYTE	0006	_DATA
@CODE	TEXT	_TEXT	
@CODESIZE	TEXT	0	
@CPU	TEXT	0101h	
@DATASIZE	TEXT	0	
@FILENAME	TEXT	lab3	
@VERSION	TEXT	510	

Microsoft (R) Macro Assembler Version 5.10
02:43:07

12/4/22

Symbols-2

109 Source Lines
109 Total Lines
31 Symbols

48046 + 459214 Bytes symbol space free

0 Warning Errors
0 Severe Errors

