

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. 0383

Пустовалова Е.М.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2021

Цель работы.

Изучить механизм работы представлений и обработки целых чисел.

Задание.

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

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

b) 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 11.

$$f2 = \begin{cases} / - (4*i+3), & \text{при } a > b \\ \backslash 6*i - 10, & \text{при } a \leq b \end{cases}$$

$$f6 = \begin{cases} / 2*(i+1) - 4, & \text{при } a > b \\ \backslash 5 - 3*(i+1), & \text{при } a \leq b \end{cases}$$

$$f5 = \begin{cases} / \min(|i1|, 6), & \text{при } k=0 \\ \backslash |i1|+|i2|, & \text{при } k \neq 0 \end{cases}$$

Числа, используемые программой вводятся сразу в файл lab3.asm. Для реализации алгоритмов использовались команда сравнения `cmp` и условные переходы. Для функций $f2$ и $f6$ условия ($a > b$ или $a \leq b$) одинаковы, поэтому их вычисление проходит в одном блоке. Сначала командой `cmp` сверяются значения a и b . С помощью команды `jle` проверяется, что $a \leq b$, и в зависимости от результата программа переходит к блоку, где рассчитываются соответствующие

значения f2 и f6. Для операций умножения использовался битовый сдвиг влево(команда shl) и сложение (команда add). Затем данное значение k сравнивается с 0 с помощью команды je, и, в зависимости от того, соблюдено ли равенство, рассчитывается нужное значение f5.

Таблица 1 – Проверка работы программы.

№	Входные данные	Значение i1	Значение i2	Результат	Комментарии
1	a = 6 b = 4 i = 3 k = -2	-15	4	19	Верно
2	a = 1 b = 2 i = 3 k = 0	8	-7	6	Верно
3	a = 5 b = 4 i = 3 k = 2	-15	4	19	Верно
4	a = 3 b = 3 i = 2 k = 0	2	-4	2	Верно

Исходный код программы находится в приложении А.

Выводы.

В ходе выполнения данной лабораторной работы была изучена работа с целыми числами и условными переходами на языке программирования Ассемблер.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ И ФАЙЛЫ ДИАГНОСТИЧЕСКИХ СООБЩЕНИЙ

Название файла: lab3.asm

```
; Стек программы
AStack SEGMENT STACK
    DW 12 DUP(?)
AStack ENDS

;Данные программы
DATA SEGMENT
;Директивы описания данных
a      DW      6
b      DW      4
i      DW      3
k      DW     -2
i1     DW      0
i2     DW      0
T      DW      0

DATA ENDS

; Код программы
CODE SEGMENT
    ASSUME CS:CODE, DS:DATA, SS:AStack

; Головная процедура
Main PROC FAR
    push DS
    sub  AX,AX
    push AX
    mov  AX,DATA
    mov  DS,AX
    mov  CX, 0

    mov  cx, i
    mov  ax, cx
    shl  cx, 1
    shl  cx, 1 ; cx = 4i
    mov  T, cx ; T = 4i
    sub  T, ax
    sub  T, ax ; T = 2i
```

```

mov bx, b
cmp a, bx

; a>b
jle f2f6
    add cx, 3
    neg cx
    mov i1, cx
    mov cx, T
    sub cx, 2
    mov i2, cx
    jmp final

; a<=b
f2f6:
    mov cx, T
    add cx, ax
    sub cx, 5
    mov T, cx
    shl cx, 1
    mov i1, cx
    neg T
    sub T, 3
    mov cx, T
    mov i2, cx

; пакет f5
final:
mov bx, k
mov ax, i1
cmp ax, 0
jge gr1
neg ax
gr1:
cmp bx, 0
je f5second
mov cx, i2
cmp cx, 0
jge gr2
neg cx
gr2:
add cx, ax

```

```

        jmp MainFinal
f5second:
        cmp ax, 6
        jge min
        mov cx, ax      ; |i1| < 6
        jmp MainFinal
min:
        mov cx, 6       ; |i1| >= 6
        jmp MainFinal
MainFinal:
        ret

Main      ENDP
CODE      ENDS
END Main

```

Название файла: lab3.lst

_Microsoft (R) Macro Assembler Version 5.10
11:08:01

12/2/21

Page 1-1

```

                                ; Стек программы
0000      AStack SEGMENT  STACK
0000  000C[                      DW 12 DUP(?)
                                ????
                                ]

0018      AStack  ENDS

                                ;Данные программы
0000      DATA      SEGMENT

                                ;Директивы описания данны
                                x
0000  0006      a      DW      6
0002  0004      b      DW      4
0004  0003      i      DW      3
0006  FFFE      k      DW     -2
0008  0000      i1     DW      0
000A  0000      i2     DW      0
000C  0000      T      DW      0

000E      DATA      ENDS

                                ; Код программы
0000      CODE      SEGMENT

                                ASSUME CS:CODE, DS:DATA, SS:AStack

                                ; Головная процедура
0000      Main      PROC  FAR
0000  1E          push  DS
0001  2B C0          sub   AX,AX
0003  50          push  AX
0004  B8 ---- R     mov   AX,DATA
0007  8E D8          mov   DS,AX
0009  B9 0000        mov   CX, 0

000C  8B 0E 0004 R   mov  cx, i
0010  8B C1          mov  ax, cx
```

0012	D1 E1	shl cx, 1
0014	D1 E1	shl cx, 1 ; cx = 4i
0016	89 0E 000C R	mov T, cx ; T = 4i
001A	29 06 000C R	sub T, ax
001E	29 06 000C R	sub T, ax ; T = 2i
0022	8B 1E 0002 R	mov bx, b
0026	39 1E 0000 R	cmp a, bx

; a>b

002A	7E 17	jle f2f6
002C	83 C1 03	add cx, 3
002F	F7 D9	neg cx
0031	89 0E 0008 R	mov i1, cx
0035	8B 0E 000C R	mov cx, T
0039	83 E9 02	sub cx, 2
003C	89 0E 000A R	mov i2, cx
0040	EB 25 90	jmp final


```
                                ; a<=b
0043                                f2f6:
0043 8B 0E 000C R                mov cx, T
0047 03 C8                      add cx, ax
0049 83 E9 05                    sub cx, 5
004C 89 0E 000C R                mov T, cx
0050 D1 E1                      shl cx, 1
0052 89 0E 0008 R                mov i1, cx
0056 F7 1E 000C R                neg T
005A 83 2E 000C R 03            sub T, 3
005F 8B 0E 000C R                mov cx, T
0063 89 0E 000A R                mov i2, cx

                                ; packet f5
0067                                final:
0067 8B 1E 0006 R                mov bx, k
006B A1 0008 R                mov ax, i1
006E 3D 0000                    cmp ax, 0
0071 7D 02                      jge gr1
0073 F7 D8                      neg ax
0075                                gr1:
0075 83 FB 00                    cmp bx, 0
0078 74 10                      je f5second
007A 8B 0E 000A R                mov cx, i2
007E 83 F9 00                    cmp cx, 0
0081 7D 02                      jge gr2
0083 F7 D9                      neg cx
0085                                gr2:
0085 03 C8                      add cx, ax
0087 EB 11 90                    jmp MainFinal
008A                                f5second:
008A 3D 0006                    cmp ax, 6
008D 7D 05                      jge min
008F 8B C8                      mov cx, ax      ; |i1| < 6
0091 EB 07 90                    jmp MainFinal
0094                                min:
0094 B9 0006                      mov cx, 6      ; |i1| >= 6
```

```
0097 EB 01 90                                jmp MainFinal
009A                                MainFinal:
009A CB                                ret

009B                                Main      ENDP
009B                                CODE      ENDS
                                END Main
```

Segments and Groups:

N a m e	Length	Align	Combine Class
ASTACK	0018	PARA	STACK
CODE	009B	PARA	NONE
DATA	000E	PARA	NONE

Symbols:

N a m e	Type	Value	Attr	
A	L WORD	0000	DATA	
B	L WORD	0002	DATA	
F2F6	L NEAR	0043	CODE	
F5SECOND	L NEAR	008A	CODE	
FINAL	L NEAR	0067	CODE	
GR1	L NEAR	0075	CODE	
GR2	L NEAR	0085	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 = 009B
MAINFINAL	L NEAR	009A	CODE	
MIN	L NEAR	0094	CODE	
T	L WORD	000C	DATA	
@CPU	TEXT	0101h		
@FILENAME	TEXT	lab3		

@VERSION TEXT 510

94 Source Lines

94 Total Lines

23 Symbols

48014 + 461293 Bytes symbol space free

0 Warning Errors

0 Severe Errors