

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:

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 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
    shl  cx, 1 ; cx = 2i
    mov  ax, cx
    add  ax, i ; ax = 3i
    mov  bx, b
    cmp  a, bx
```

```

; a>b
jle f2f6
    sub cx, 2
    mov i2, cx
    shl cx, 1
    neg cx
    sub cx, 7
    mov i1, cx
    jmp final

; a<=b
f2f6:
    mov cx, ax
    sub cx, 2
    neg cx
    mov i2, cx
    neg cx
    shl cx, 1
    sub cx, 6
    mov i1, cx

; расчет f5
final:
mov ax, i1
cmp ax, 0
jge gr1
neg ax
gr1:
mov bx, k
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
 01:07:54

12/9/21

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  D1 E1              shl  cx, 1 ; cx = 2i
0012  8B C1              mov  ax, cx
0014  03 06 0004 R      add  ax, i ; ax = 3i
0018  8B 1E 0002 R      mov  bx, b
001C  39 1E 0000 R      cmp  a, bx

; a>b
0020  7E 15              jle  f2f6
0022  83 E9 02              sub  cx, 2
0025  89 0E 000A R      mov  i2, cx
0029  D1 E1              shl  cx, 1
002B  F7 D9              neg  cx
002D  83 E9 07              sub  cx, 7
0030  89 0E 0008 R      mov  i1, cx
0034  EB 17 90              jmp  final

; a<=b
0037      f2f6:
0037  8B C8              mov  cx, ax

```

```
0039 83 E9 02                sub cx, 2
003C F7 D9                  neg cx
003E 89 0E 000A R          mov i2, cx
0042 F7 D9                  neg cx
0044 D1 E1                  shl cx, 1
0046 83 E9 06              sub cx, 6
0049 89 0E 0008 R          mov i1, cx

                                ; расчет f5
004D                        final:
004D A1 0008 R              mov ax, i1
0050 3D 0000                cmp ax, 0
0053 7D 02                  jge gr1
0055 F7 D8                  neg ax
0057                        gr1:
0057 8B 1E 0006 R          mov bx, k
005B 83 FB 00                cmp bx, 0
005E 74 10                  je f5second
0060 8B 0E 000A R          mov cx, i2
0064 83 F9 00                cmp cx, 0
0067 7D 02                  jge gr2
0069 F7 D9                  neg cx
006B                        gr2:
006B 03 C8                  add cx, ax
006D EB 11 90                jmp MainFinal
0070                        f5second:
0070 3D 0006                cmp ax, 6
0073 7D 05                  jge min
0075 8B C8                  mov cx, ax      ; |i1| < 6
0077 EB 07 90                jmp MainFinal
007A                        min:
007A B9 0006                mov cx, 6      ; |i1| >= 6
007D EB 01 90                jmp MainFinal
0080                        MainFinal:
0080 CB                      ret

0081                        Main      ENDP
```


0081

CODE

ENDS

END Main

Segments and Groups:

N a m e	Length	Align	Combine Class
ASTACK	0018	PARA	STACK
CODE	0081	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	0037	CODE	
F5SECOND	L NEAR	0070	CODE	
FINAL	L NEAR	004D	CODE	
GR1	L NEAR	0057	CODE	
GR2	L NEAR	006B	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 = 0081
MAINFINAL	L NEAR	0080	CODE	
MIN	L NEAR	007A	CODE	
T	L WORD	000C	DATA	
@CPU	TEXT	0101h		
@FILENAME	TEXT	lab3		

@VERSION TEXT 510

89 Source Lines

89 Total Lines

23 Symbols

48014 + 461293 Bytes symbol space free

0 Warning Errors

0 Severe Errors

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