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. 0383 Куликов А. В.

Преподаватель Ефремов М. А.

Санкт-Петербург

2021

Цель работы.

Изучить представление и обработку целых чисел, организацию ветвящихся процессов на языке Ассемблера.

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 функции f1, f2, f3:

$$f1 = \begin{cases} -(4*i+3), & \text{при } a>b \\ 6*i-10, & \text{при } a<=b \end{cases}$$

$$f2 = \begin{cases} -(6*i+8), & \text{при } a>b \\ 9-3*(i-1), & \text{при } a<=b \end{cases}$$

$$f3 = \begin{cases} |i1 + i2|, & \text{при } k=0 \\ \min(i1, i2), & \text{при } k/=0 \end{cases}$$

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.

В ходе выполнения работы были использованы инструкции cmp, jle, jge, jl, jg, jne, je, использованные для реализации ветвления и сравнения веденных чисел.

Все возможные варианты работы кода программы приведены в Табл.1.

Табл.1

Значения a, b, i, k	Результат вычисления i1	Результат вычисления i2	Результат вычисления res	Прим.
a = 1 b = 0 i = -5 k = 0	11 (hex) = 17 (dec)	16 (hex) = 22 (dec)	27 (hex) = 39 (dec)	Верно
a = 1 b = 0 i = -5 k = 1	11 (hex) = 17 (dec)	16 (hex) = 22 (dec)	11 (hex) = 17 (dec)	Верно
a = -1 b = 10 i = 2 k = 0	2 (hex) = 2 (dec)	6 (hex) = 6 (dec)	8 (hex) = 8 (dec)	Верно
a = -1 b = -1 i = -2 k = -7	FFEA(hex) = -22 (dec)	12 (hex) = 18 (dec)	FFEA (hex) = -22 (dec)	Верно

Выводы.

В ходе лабораторной работы были изучены представление и обработка целых чисел и организация ветвящихся процессов в языке Ассемблера.

ПРИЛОЖЕНИЕ А

Тексты исходных файлов программ

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
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 DX, AX mov a,
```

```
        0 mov b, 0 mov i, 0
                mov k, 0


        mov cx, a
sub cx, b       cmp
cx, 0   jle L1


        ;a > b
        ;i1 = -(4 * i + 3)
mov cx, i       shl cx, 1
shl cx, 1       add cx, 3
neg cx          mov i1,
cx


        ;i2 = -(6 * i + 8)
mov cx, i       shl cx, 1
add cx, i       add cx, 4
shl cx, 1 neg cx mov
i2, cx


        jmp L2
        ;a <= b L1:


        ;i1 = 6 * i - 10
        mov cx, i
shl cx, 1       add cx,
i       shl cx, 1
sub cx, 10      mov i1,
cx
```

```
        ;i2 = 9 - 3 * (i - 1)
        mov cx, i
shl cx, 1       add
cx, i   neg cx
add cx, 12
        mov i2, cx

        L2:
        cmp k, 0
jne L3

        ;k == 0
        cmp i1, 0
        jl i10
        cmp i2, 0
        jl i20
        jmp endf3

        i10:
        neg i1
        cmp i2, 0
jg endf3

        i20:
neg i2

        endf3:
mov ax, i1   add
```

```
ax, i2          jmp
endmain

        ;k != 0
        L3:
        mov cx, i1
sub cx, i2     cmp
cx, 0  jl Li1
mov AX, i2
jmp endmain

        Li1:    mov
AX, i1 jmp
endmain

        endmain:
        ret
Main ENDP
CODE ENDS

END Main
```

ПРИЛОЖЕНИЕ Б

Тексты файлов диагностических сообщений программ

lab3.lst

```
MICROSOFT (R) MACRO ASSEMBLER VERSION 5.10                11/25/21
04:25:1
                                        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                    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  8B D0                            MOV DX, AX
```

```
0009  C7 06 0000 R 0000          MOV A, 0
000F  C7 06 0002 R 0000          MOV B, 0
0015  C7 06 0004 R 0000          MOV I, 0
001B  C7 06 0006 R 0000          MOV K, 0

0021  8B 0E 0000 R                MOV CX, A

0025  2B 0E 0002 R                SUB CX, B

0029  83 F9 00                    CMP CX, 0

002C  7E 29                       JLE L1

                                ;A > B
                                ;I1 = -(4 * I + 3)
002E  8B 0E 0004 R                  MOV CX, I
0032  D1 E1                       SHL CX, 1
0034  D1 E1                       SHL CX, 1
0036  83 C1 03                      ADD CX, 3
0039  F7 D9                       NEG CX
003B  89 0E 0008 R                 MOV I1, CX

                                ;I2 = -(6 * I + 8)
003F  8B 0E 0004 R                  MOV CX, I
0043  D1 E1                       SHL CX, 1
0045  03 0E 0004 R                ADD CX, I
0049  83 C1 04                      ADD CX, 4
004C  D1 E1                       SHL CX, 1
MICROSOFT (R) MACRO ASSEMBLER VERSION 5.10          11/25/21
04:25:1
                                PAGE    1-2


004E  F7 D9                       NEG CX

0050  89 0E 000A R                MOV I2, CX


0054  EB 27 90                    JMP L2
```

```
                                ;A <= B
0057                            L1:

                                ;I1 = 6 * I - 10
0057  8B 0E 0004 R                  MOV CX, I
005B  D1 E1                         SHL CX, 1
005D  03 0E 0004 R                  ADD CX, I
0061  D1 E1                         SHL CX, 1
0063  83 E9 0A                      SUB CX, 10
0066  89 0E 0008 R                  MOV I1, CX

                                ;I2 = 9 - 3 * (I - 1)
006A  8B 0E 0004 R                  MOV CX, I
006E  D1 E1                         SHL CX, 1
0070  03 0E 0004 R                  ADD CX, I
0074  F7 D9                         NEG CX
0076  83 C1 0C                      ADD CX, 12
0079  89 0E 000A R                  MOV I2, CX

007D                                L2:
007D  83 3E 0006 R 00                   CMP K, 0
0082  75 2A                         JNE L3

                                ;K == 0
0084  83 3E 0008 R 00               CMP I1, 0
0089  7C 0A                         JL I10
008B  83 3E 000A R 00                   CMP I2, 0
0090  7C 0E                         JL I20
0092  EB 10 90                      JMP ENDF3

0095                            I10:
0095  F7 1E 0008 R                  NEG I1
0099  83 3E 000A R 00                   CMP I2, 0
009E  7F 04                         JG ENDF3
```

```
00A0                            I20:
00A0  F7 1E 000A R              NEG I2

00A4                            ENDF3:
00A4  A1 0008 R                 MOV AX, I1
00A7  03 06 000A R              ADD AX, I2
00AB  EB 1A 90                  JMP ENDMAIN

                        ;K != 0
00AE                            L3:
00AE  8B 0E 0008 R              MOV CX, I1
00B2  2B 0E 000A R              SUB CX, I2
00B6  83 F9 00                  CMP CX, 0
```



```
00B9  7C 06                     JL LI1
00BB  A1 000A R                 MOV AX, I2
00BE  EB 07 90                  JMP ENDMAIN

00C1                    LI1:
00C1  A1 0008 R         MOV AX, I1
00C4  EB 01 90                  JMP ENDMAIN

00C7                    ENDMAIN:
00C7  CB                RET
00C8                    MAIN ENDP
00C8                    CODE ENDS

                        END MAIN
```


SEGMENTS AND GROUPS:

N A M E	LENGTH	ALIGN	COMBINE	CLASS

```
ASTACK . . . . . .      0018 PARA       STACK
CODE . . . . . .        00C8 PARA       NONE
DATA . . . . . .        000C PARA       NONE

SYMBOLS:

            N A M E             TYPE        VALUE     ATTR

A . . . . . .  L WORD      0000 DATA

B . . . . . .  L WORD      0002 DATA

ENDF3 . . . . . .         L NEAR     00A4 CODE
ENDMAIN . . . . . .   L NEAR         00C7 CODE

I . . . . . .  L WORD      0004 DATA
I1 . . . . . .  L WORD     0008 DATA
I10 . . . . . .  L NEAR    0095 CODE
I2 . . . . . .  L WORD     000A DATA
I20 . . . . . .  L NEAR    00A0 CODE

K . . . . . .  L WORD      0006 DATA

L1 . . . . . .  L NEAR     0057 CODE
L2 . . . . . .  L NEAR     007D CODE
L3 . . . . . .  L NEAR     00AE CODE
LI1 . . . . . .  L NEAR    00C1 CODE

MAIN . . . . . .          F PROC      0000 CODE       LENGTH = 00C8

@CPU . . . . . .          TEXT  0101H
@FILENAME . . . . . .           TEXT  LAB3
@VERSION . . . . . .   TEXT  510


   119 SOURCE  LINES
   119 TOTAL   LINES
    23 SYMBOLS

 48058 + 461249 BYTES SYMBOL SPACE FREE
```

0 WARNING ERRORS
0 SEVERE ERRORS