

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

Смирнов И.А.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2021

Цель работы.

Познакомиться с представлением и обработкой целых чисел на языке ассемблер.

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 .

Замечания:

1) при разработке программы нельзя использовать фрагменты, представленные на ЯВУ, в частности, для ввода-вывода данных. Исходные данные должны вводиться, а результаты контролироваться в режиме отладки;

2) при вычислении функций $f1$ и $f2$ вместо операции умножения следует использовать арифметический сдвиг и, возможно, сложение;

3) при вычислении функций $f1$ и $f2$ нельзя использовать процедуры;

4) при разработке программы следует минимизировать длину кода, для чего, если надо, следует преобразовать исходные выражения для вычисления функций.

Вариант 15 (шифр задания – 3.5.3):

$/7-4*i$, при $a>b$

$i1 = <$

$\backslash 8 - 6*i$, при $a \leq b$

$/20-4*i$, при $a>b$

$i2 = <$

$\backslash -(6*i - 6)$, при $a \leq b$

$/ |i1+i2|$, при $k = 0$

$res = <$

$\backslash \min(i1, i2)$, при $k \neq 0$

Выполнение работы.

Для расчета значений функций использовались команды `cmp`, `jne`, `jle`, `jl`.

Для сложения использовалась команда `add`, а для умножения – битовый сдвиг влево `shl`.

Разработанный программный код см. в приложении А.

Текст файла диагностических сообщений см. в приложении Б.

Тестирование.

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

Значения	i1	i2	res	Комментарий
a = 2 b = 1 k = 0 i = 1	0003h = 3	0010h = 16	0013h = 19	ВЕРНО
a = 2	0003h = 3	0010h = 16	0003h = 3	ВЕРНО

b = 1 k = 1 i = 1				
a = 1 b = 2 k = 0 i = -2	0014h = 20	0012h = 18	0026h = 38	ВЕРНО
a = 1 b = 2 k = 1 i = -2	0014h = 20	0012h = 18	0012h = 18	ВЕРНО

Выводы.

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

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lr3.asm

```
AStack SEGMENT STACK
    DW 12 DUP(?)
AStack ENDS

;Данные программы
DATA SEGMENT
;Директивы описания данных
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

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

    mov    a, 1
    mov    b, 2
    mov    i, -2
    mov    k, 0

    ;вычисление f3
    mov    cx, i
    shl    cx, 1
    mov    ax, cx
    shl    cx, 1
    mov    bx, b    ;
    cmp    a, bx    ; сравнение a и b
    jle    flsecond
    neg    cx
    add    cx, 7
    jmp    flfinal
flsecond:
    add    cx, ax
    neg    cx
    add    cx, 8
flfinal:
```

```

mov i1, cx

;вычисление f5
mov cx, i
shl cx, 1
mov ax, cx
shl cx, 1
cmp a, bx
jle f2second
    neg cx
    add cx, 20
    jmp f2final
f2second:
    add cx, ax
    add cx, -6
    neg cx
f2final:
mov i2, cx

;рассчет res
mov bx, k
cmp bx, 0
jne resSecond
    mov bx, i1
    add bx, i2
    cmp bx, 0
    jge skip
    neg bx
    skip:
    mov cx, bx
    jmp MainFinal
resSecond:
    mov bx, i1
    cmp bx, i2
    jl min
    mov cx, i2
    jmp MainFinal
min:
    mov cx, bx
    jmp MainFinal
MainFinal:                ; в cx лежит значение res
ret
Main        ENDP
CODE        ENDS
END Main

```

ПРИЛОЖЕНИЕ Б

ФАЙЛ ДИАГНОСТИЧЕСКИХ СООБЩЕНИЙ

Название файла: lr3.lst

Microsoft (R) Macro Assembler Version 5.10
17:02:1

11/15/21

Page

1-1

```

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

0018          AStack  ENDS

;Данные программы
0000          DATA          SEGMENT

;Директивы описания данных
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  8E D8          mov   DS,AX
0009  B9 0000          mov   CX, 0

000C  C7 06 0000 R 0001          mov a, 1
0012  C7 06 0002 R 0002          mov b, 2
0018  C7 06 0004 R FFFE          mov i, -2
001E  C7 06 0006 R 0000          mov k, 0

;вычисление f3
0024  8B 0E 0004 R          mov cx, i
0028  D1 E1          shl  cx, 1
002A  8B C1          mov  ax, cx
002C  D1 E1          shl  cx, 1
002E  8B 1E 0002 R          mov bx, b      ;
0032  39 1E 0000 R          cmp  a, bx      ; сравнение a и b

```

```

0036 7E 08                jle flsecond
0038 F7 D9                neg cx
003A 83 C1 07            add cx, 7
003D EB 08 90            jmp flfinal
0040                    flsecond:
0040 03 C8                add cx, ax
0042 F7 D9                neg cx
0044 83 C1 08            add cx, 8
0047                    flfinal:
0047 89 0E 0008 R        mov i1, cx

```

Microsoft (R) Macro Assembler Version 5.10
17:02:1

11/15/21

Page

1-2

```

                                ;вычисление f5
004B 8B 0E 0004 R        mov cx, i
004F D1 E1                shl cx, 1
0051 8B C1                mov ax, cx
0053 D1 E1                shl cx, 1
0055 39 1E 0000 R        cmp a, bx
0059 7E 08                jle f2second
005B F7 D9                neg cx
005D 83 C1 14            add cx, 20
0060 EB 08 90            jmp f2final
0063                    f2second:
0063 03 C8                add cx, ax
0065 83 C1 FA            add cx, -6
0068 F7 D9                neg cx
006A                    f2final:
006A 89 0E 000A R        mov i2, cx

                                ;рассчет res
006E 8B 1E 0006 R        mov bx, k
0072 83 FB 00            cmp bx, 0
0075 75 14                jne resSecond
0077 8B 1E 0008 R        mov bx, i1
007B 03 1E 000A R        add bx, i2
007F 83 FB 00            cmp bx, 0
0082 7D 02                jge skip
0084 F7 DB                neg bx
0086                    skip:
0086 8B CB                mov cx, bx
0088 EB 17 90            jmp MainFinal
008B                    resSecond:
008B 8B 1E 0008 R        mov bx, i1
008F 3B 1E 000A R        cmp bx, i2
0093 7C 07                jl min
0095 8B 0E 000A R        mov cx, i2
0099 EB 06 90            jmp MainFinal
009C                    min:
009C 8B CB                mov cx, bx
009E EB 01 90            jmp MainFinal

```



```

00A1                                MainFinal:                ; в сx лежит зна
                                чение res
00A1    CB                        ret
00A2                                Main        ENDP
00A2                                CODE        ENDS
                                END Main

```

Microsoft (R) Macro Assembler Version 5.10
17:02:1

11/15/21

Symbols-1

Segments and Groups:

	N a m e	Length	Align	Combine	Class
ASTACK	0018	PARA	STACK	
CODE	00A2	PARA	NONE	
DATA	000C	PARA	NONE	

Symbols:

	N a m e	Type	Value	Attr
A	L WORD	0000	DATA
B	L WORD	0002	DATA
F1FINAL	L NEAR	0047	CODE
F1SECOND	L NEAR	0040	CODE
F2FINAL	L NEAR	006A	CODE
F2SECOND	L NEAR	0063	CODE
I	L WORD	0004	DATA
I1	L WORD	0008	DATA
I2	L WORD	000A	DATA
K	L WORD	0006	DATA
00A2 MAIN	F PROC	0000	CODE Length =
MAINFINAL	L NEAR	00A1	CODE
MIN	L NEAR	009C	CODE
RESSECOND	L NEAR	008B	CODE
SKIP	L NEAR	0086	CODE
@CPU	TEXT	0101h	
@FILENAME	TEXT	LR3	
@VERSION	TEXT	510	

94 Source Lines
94 Total Lines

23 Symbols

47996 + 461311 Bytes symbol space free

0 Warning Errors

0 Severe Errors