

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: Программирование ветвящихся процессов.

Студент гр. 0383

Пенкин М.В.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2021

Цель работы.

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 8:

$/ - (4*i+3)$, при $a>b$

$i1 = f2 = <$

$\backslash 6*i -10$, при $a\leq b$

$/ 7 - 4*i$, при $a > b$

$i2 = f3 = <$

$\backslash 8 - 6*i$, при $a \leq b$

$/ |i1| - |i2|$, при $k < 0$

$res = f7 = <$

$\backslash \max(4, |i2| - 3)$, при $k \geq 0$

Выполнение работы.

Таблица 1. Проверка работы программы с помощью отладчика

Введенные значения a, b, i, k	Полученное значение i1	Полученное значение i2	Полученное значение res	Комментарий
a = -6 b = -2 i = 5 k = -10	0014h = 20	FFEA = -22	FFFEh = -2	Верно, $a < b \Rightarrow$ f2 = $6*i - 10 = 20$, f3 = $8 - 6*i = -22$, $k < 0 \Rightarrow$ f8 = $ i1 - i2 = -2$
a = 30 b = 20 i = -10 k = 10	0025h = 37	002Fh = 47	002C = 44	Верно, $a > b \Rightarrow$ f2 = $-(4*i + 3) = 37$, f3 = $7 - 4*i = 47$, $k > 0 \Rightarrow$ f8 = $\max(4, i2 - 3) = 44$
a = -3 b = -3 i = -3 k = -3	FFE4h = -28	001Ah = 26	0002h = 2	Верно, $a = b \Rightarrow$ f2 = $6*i - 10 = -28$, f3 = $8 - 6*i = 26$, $k < 0 \Rightarrow$ • f8 = $ i1 - i2 = 2$
a = -20 b = 30 i = 1 k = 5	FFFC = -4	0002h = 2	0004h = 4	Верно, $a < b \Rightarrow$ f2 = $6*i - 10 = -4$, f3 = $8 - 6*i = 2$, $k > 0 \Rightarrow$ f8 = $\max(4, i2 - 3) = 4$

Тексты исходных файлов программ см. в приложении А.

Тексты файлов диагностических сообщений см. в приложении Б.

Выводы.

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

ПРИЛОЖЕНИЕ А

ТЕКСТЫ ИСХОДНЫХ ФАЙЛОВ ПРОГРАММ

Название файла: LR3.asm

```
; Стек программы
AStack SEGMENT STACK
    DW 12 DUP(?)
AStack ENDS
;Данные программы
DATA SEGMENT
;Директивы описания данных
a      DW      -6
b      DW      -2
i      DW      5
k      DW     -10
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    DS,AX
    mov    CX, 0

    ;вычисление f2
    mov    cx, i
    mov    ax, cx
    shl    cx, 1
    shl    cx, 1
    mov    bx, b      ;
    cmp    a, bx      ; сравнение a и b
    jle    f2second
    add    cx, 3
    neg    cx
    jmp    f2final
f2second:
    add    cx, ax
    add    cx, ax
    add    cx, -10
f2final:
    mov    i1, cx

    ;вычисление f3
    mov    cx, i
    mov    ax, cx
```

```

shl cx, 1
shl cx, 1
cmp a, bx
jle f3second
    mov ax, cx
    mov cx, 7
    sub cx, ax
    jmp f3final
f3second:
    add cx, ax
    add cx, ax
    mov ax, cx
    mov cx, 8
    sub cx, ax
f3final:
mov i2, cx

mov cx, i1
cmp cx, 0
jge skip1      ;модуль i1
    neg cx
    mov i1, cx
skip1:

mov cx, i2
cmp cx, 0
jge skip2      ;модуль i2
    neg cx
    mov i2, cx
skip2:

;расчет f8
mov bx, k
cmp bx, 0
jl f8Second
    mov bx, i2
    sub bx, 3
    cmp bx, 4
    jl max1
    mov cx, bx      ; |i1| >= 4
    jmp MainFinal
max1:
    mov cx, 4      ; |i1| < 4
    jmp MainFinal
f8Second:
    mov cx, i1
    sub cx, i2
MainFinal:      ; в cx лежит значение функции f8
    ret
Main          ENDP
CODE          ENDS
END Main

```



```

; 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

;вычисление f2
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
0016  8B 1E 0002 R    mov  bx, b ;
001A  39 1E 0000 R    cmp  a, bx ; сравнение a й
, b
001E  7E 08          jle  f2second
0020  83 C1 03        add  cx, 3
0023  F7 D9          neg  cx
0025  EB 08 90        jmp  f2final
0028                f2second:
0028  03 C8          add  cx, ax
002A  03 C8          add  cx, ax
002C  83 C1 F6        add  cx, -10
002F                f2final:
002F  89 0E 0008 R    mov  i1, cx

;вычисление f3
0033  8B 0E 0004 R    mov  cx, i
Microsoft      (R)      Macro      Assembler      Version      5.10
10/19/21 12:26:5
Page
1-2

0037  8B C1          mov  ax, cx
0039  D1 E1          shl  cx, 1
003B  D1 E1          shl  cx, 1
003D  39 1E 0000 R    cmp  a, bx
0041  7E 0A          jle  f3second
0043  8B C1          mov  ax, cx

```



```

0045 B9 0007          mov cx, 7
0048 2B C8          sub cx, ax
004A EB 0C 90        jmp f3final
004D                f3second:
004D 03 C8          add cx, ax
004F 03 C8          add cx, ax
0051 8B C1          mov ax, cx
0053 B9 0008          mov cx, 8
0056 2B C8          sub cx, ax
0058                f3final:
0058 89 0E 000A R      mov i2, cx

005C 8B 0E 0008 R      mov cx, i1
0060 83 F9 00        cmp cx, 0
0063 7D 06          jge skip1      ;модуль i1
0065 F7 D9          neg cx
0067 89 0E 0008 R      mov i1, cx
006B                skip1:

006B 8B 0E 000A R      mov cx, i2
006F 83 F9 00        cmp cx, 0
0072 7D 06          jge skip2      ;модуль i2
0074 F7 D9          neg cx
0076 89 0E 000A R      mov i2, cx
007A                skip2:

                                ;расчет f8
007A 8B 1E 0006 R      mov bx, k
007E 83 FB 00        cmp bx, 0
0081 7C 17          jl f8Second
0083 8B 1E 000A R      mov bx, i2
0087 83 EB 03        sub bx, 3
008A 83 FB 04        cmp bx, 4
008D 7C 05          jl max1
008F 8B CB          mov cx, bx      ; |i1| >=
4                                4
0091 EB 0F 90        jmp MainFinal
0094                max1:
0094 B9 0004          mov cx, 4      ; |i1| < 4
0097 EB 09 90        jmp MainFinal
009A                f8Second:
009A 8B 0E 0008 R      mov cx, i1

```

```

009E  2B 0E 000A R                               sub cx, i2
00A2                                     MainFinal:           ; в cx лежи
                                     т значение функции f8
00A2  CB                               ret
00A3                                     Main      ENDP
00A3                                     CODE      ENDS
                                     END Main
Microsoft      (R)      Macro      Assembler      Version      5.10
10/19/21 12:26:5

```

Symbols-1

Segments and Groups:

Class	N a m e	Length	Align	Combine
ASTACK	0018	PARA	STACK
CODE	00A3	PARA	NONE
DATA	000C	PARA	NONE

Symbols:

	N a m e	Type	Value	Attr
A	L WORD	0000	DATA
B	L WORD	0002	DATA
F2FINAL	L NEAR	002F	CODE
F2SECOND	L NEAR	0028	CODE
F3FINAL	L NEAR	0058	CODE
F3SECOND	L NEAR	004D	CODE
F8SECOND	L NEAR	009A	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 =
00A3				
MAINFINAL	L NEAR	00A2	CODE	
MAX1	L NEAR	0094	CODE	
SKIP1	L NEAR	006B	CODE	
SKIP2	L NEAR	007A	CODE	
@CPU	TEXT	0101h		
@FILENAME	TEXT	LR3		
@VERSION	TEXT	510		

102 Source Lines
102 Total Lines
24 Symbols

47992 + 461315 Bytes symbol space free

0 Warning Errors
0 Severe Errors