

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
ВАРИАНТ 14

Студент гр. 1383

Манучарова А.С.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

Цель работы.

Изучить организацию ветвящихся процессов и разработать программу 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 .

Выполнение работы.

Были произведены следующие упрощения:

1. $8 - 6*i = 8 - (2*i + i)*2$
2. $-(6*i - 4) = -((2*i+i)*2 - 4)$
3. $3*(i+2) = 2*(i+2) + (i+2)$

При $a > b$ значение i в функциях $f1$, $f2$ увеличено как минимум в два раза, поэтому сразу выполняется операция $i*2$.

В $f3$ можно до сравнения значения k с нулем сделать значение $i2$ отрицательным.

Результаты выполнения функций записываются в ячейки памяти $[f1]$, $[f2]$, $[f3]$. Результаты работы программы представлены в таблице 1.

Таблица 1. Протокол example

№	Исходные данные	Вывод программы	Комментарий
1	a=6, b=3, i=2, k=1	FFFF ₁₆ = -1 ₁₀ FFF8 = -8 0007 ₁₆ = 7 ₁₀	[f1] = FFFF [f2] = F8FF [f3] = 07 00
2	a=3, b=6, i=5, k=-1	FFEA ₁₆ = -22 ₁₀ 0015 ₁₆ = 21 ₁₀ FFF5 ₁₆ = -11 ₁₀	[f1] = EAFF [f2] = 15 00 [f3] = F5FF
3	a = 2, b = 2, i = -1, k = 0	000E ₁₆ = 14 ₁₀ 0003 ₁₆ = 3 ₁₀ 000B ₁₆ = 11 ₁₀	[f1] = 0E00 [f2] = 03 00 [f3] = 0B00
4	a = -4, b = 0, i = -2, k = 1	0014 ₁₆ = 20 ₁₀ 0000 ₁₆ = 0 ₁₀ 0014 ₁₆ = 20 ₁₀	[f1] = 14 00 [f2] = 00 00 [f3] = 14 00
5	a = -4, b = -7, i = -2, k = -2	000F ₁₆ = 15 ₁₀ 0010 ₁₆ = 16 ₁₀ 000F ₁₆ = 15 ₁₀	[f1] = 0F 00 [f2] = 10 00 [f3] = 0F 00
6	a = -4, b = -7, i = -2, k = 1	000F ₁₆ = 15 ₁₀ 0010 ₁₆ = 16 ₁₀ 0001 ₁₆ = 1 ₁₀	[f1] = 0F 00 [f2] = 10 00 [f3] = 01 00
7	a = -4, b = -7, i = 3, k = 1	FFFB ₁₆ = -5 ₁₀ FFF2 ₁₆ = -14 ₁₀ 0009 ₁₆ = 9 ₁₀	[f1] = FBFF [f2] = F2FF [f3] = 09 00

Выводы.

В ходе выполнения работы была изучена организация ветвящихся процессов и разработана эффективная программа на языке Ассемблера.

ПРИЛОЖЕНИЕ А

КОД ПРОГРАММЫ

example.asm:

```
;          / 7 - 4*i , при a>b
;  (f1)f3 = <
;          \ 8 -6*i , при a<=b

;          / -(6*i - 4) , при a>b
;  (f2)f4 = <
;          \ 3*(i+2) , при a<=b

;          / max(i1,10-i2), при k<0
;  (f3)f2 = <
;          \ |i1 - i2| , при k>=0

.Model small
a EQU -4
b EQU -7
i EQU 3
k EQU 1

.STACK

.data
f1 DW 0
f2 DW 0
f3 DW 0
```

```

.CODE

    mov cx, i

    mov ax, a
    mov bx, b
    cmp ax, bx
    jle less

    shl cx, 1
    mov dx, cx

    shl cx, 1
    neg cx
    add cx, 7

    add dx, i
    shl dx, 1
    sub dx, 4
    neg dx

    jmp final1

less:
    mov dx, cx

    shl cx, 1
    add cx, i
    shl cx, 1
    neg cx
    add cx, 8

```

```

    add dx, 2
    mov ax, dx
    shl dx, 1
    add dx, ax

final1:
    mov [f1], cx
    mov [f2], dx

    neg dx
    mov bx, k
    cmp bx, 0
    jge greater

    add dx, 10
    cmp cx, dx
    jg final2
    xchg cx, dx
    jmp final2

greater:
    add cx, dx
    jns final2
    neg cx

final2:
    mov [f3], cx
END

```