

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ
по лабораторной работе №3
по дисциплине «Представление и обработка целых чисел. Организация
ветвящихся процессов»
Тема: Трансляции, отладка и выполнение программ на языке
Ассемблера
Вариант 1.2.1

Студент гр. 1383

Богданов Е.М.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

Цели работы.

Разработать на языке Ассемблера программу, которая по заданным целочисленным значениям параметров вычисляет значения функций.

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 .

$$\begin{aligned} f1 &= \begin{cases} / 15-2*i, & \text{при } a>b \\ \backslash 3*i+4, & \text{при } a\leq b \end{cases} \\ f2 &= \begin{cases} / -(4*i+3), & \text{при } a>b \\ \backslash 6*i-10, & \text{при } a\leq b \end{cases} \\ f3 &= \begin{cases} / 7-4*i, & \text{при } a>b \\ \backslash \min(i1,i2), & \text{при } k=0 \\ \backslash \max(i1,i2), & \text{при } k\neq 0 \end{cases} \end{aligned}$$

Выполнение работы.

Были реализованы функции из Каталога Заданий, соответствующие 1 варианту. Реализованная программа протранслирована с различными тестовыми данными.

Для выполнения данного задания были использованы такие команды:

mov – присваивание

add — сложение

sub — вычитание

cmp – сравнение

neg – смена знака

sal — арифметический сдвиг влево

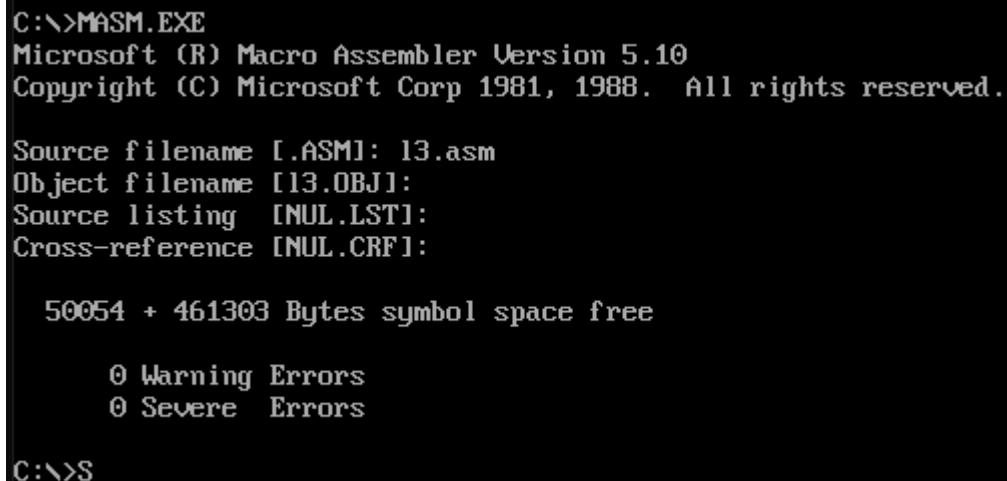
jmp – команда безусловного перехода

jg(jump greater) — выполняет короткий переход, если первый операнд больше второго операнда при выполнении операции сравнения с помощью команды cmp.

jge(jump greater equal) — выполняет короткий переход, если первый операнд меньше второго операнда при выполнении операции сравнения с помощью команды cmp.

Для реализации ветвления в программе использовались метки. Метка - это символьное имя, обозначающее ячейку памяти, которая содержит некоторую команду.

Рисунок 1 — Транслирование программы



```
C:\>MASM.EXE
Microsoft (R) Macro Assembler Version 5.10
Copyright (C) Microsoft Corp 1981, 1988. All rights reserved.

Source filename [.ASM]: 13.asm
Object filename [13.OBJ]:
Source listing [NUL.LST]:
Cross-reference [NUL.CRF]:

50054 + 461303 Bytes symbol space free

0 Warning Errors
0 Severe Errors

C:\>S
```

Выводы.

В ходе лабораторной работы были получены навыки разработки программы с заданными целочисленными значениями на языке программирования Ассемблер.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

```
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
res dw 0
DATA ENDS

ASSUME CS:CODE, SS:AStack, DS:DATA

CODE SEGMENT
Main PROC FAR
    push DS
    sub AX,AX
    push AX
    mov AX, DATA
    mov DS, AX

    mov ax,a
    cmp ax,b
    jg A_Great_B

    mov dx,i
    mov i1,dx
    sal i1,1
    mov dx,i
    add i1,dx

    mov dx,i
    mov i2,dx
    sal i2,1
    sal i2,1
```

```

    add dx,i
    add dx,i
    add dx,-10
    add i2,dx
    jmp FINAL

A_Great_B:
    mov dx,i
    mov i1,dx
    sal i1,1
    neg i1
    add i1,15

    mov dx,i
    mov i1,dx
    sal i2,1
    sal i2,1
    add i2,3
    neg i2

FINAL:
    mov ax,i1
    mov bx,i2
    cmp ax,bx
    jge NOT_SWITCH

    mov cx,ax
    mov ax,bx
    mov bx,cx

NOT_SWITCH:
    mov cx,k
    cmp cx,0
    jne MX
    mov ax,bx

MX:
    mov res,ax

Main ENDP
CODE ENDS
END Main

```