**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 |  | Желнин М.Ю. |
| Преподаватель |  | Ефремов М.А. |

Санкт-Петербург

2021

## Цель работы.

Изучить механизм работы представлений и обработки целых чисел.

## Задание.

Разработать на языке Ассемблера программу, которая по заданным целочисленным значениям параметров a, b, i, k вычисляет:

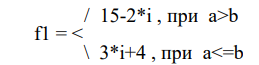
а) значения функций i1 = f1(a,b,i) и i2 = f2(a,b,i);

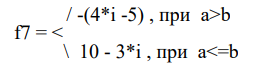
b) значения результирующей функции res = f3(i1,i2,k),

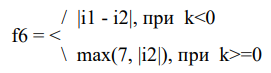
где вид функций f1 и f2 определяется из табл. 2, а функции f3 - из табл.3 по цифрам шифра индивидуального задания (n1,n2,n3), приведенным в табл.4.

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.

**Вариант 6**







## Выполнение работы.

Происходит расчет функций f1, f2, f3. При организации ветвящихся процессов использовалась функция cmp и условные переходы. Для операций умножения использовался побитовый сдвиг влево и сложение.

Разработанный программный код см. в приложении А.

## Тестирование.

Результаты тестирования представлены в табл. 1.

Таблица 1 – Проверка работы программы с помощью отладчика (все результаты заносились в реестр CX)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № п/п | Входные данные | Полученное значение i1 | Полученное значение i2 | Полученное значение res | Комментарии |
|  | a = 1  b = 2  i = 3  k = 4 | 000Dh = 13 | 0001h = 1 | 0007h = 7 | Верно, т.к. a <=b, 3\*3 + 4 = 13, (10 – 3\*3) = 1, k >= 0, max(7, 1) = 7 |
|  | a = 1  b = 2  i = 3  k = -1 | 000Fh = 13 | 0003h = 1 | 000Сh = 12 | Верно, т.к. a <=b, 3\*3 + 4 = 13, (10 – 3\*3) = 1, k < 0, |i1-i2| = 12 |
|  | a = 2  b = 1  i = 3  k = 0 | 0009h = 9 | FFF9h = -7 | 0007h = 7 | Верно, т.к. a>b, -(15 – 2\*3) = 9, -(4\*3-5) = -7, k = 1, max(7, |-7|) = 7 |
| 4 | a = 2  b = 1  i = 3  k = -1 | 0009h = 9 | FFF9h = -7 | 0010h = 16 | Верно, т.к. a>b, -(15 – 2\*3) = 9, -(4\*3-5) = -7, k = -11, |9+7| = 16 |

## Выводы.

Был изучен механизм работы представлений и обработки целых чисел.

# Приложение А Исходный код программы

Название файла: hello3.asm

; Стек программы

AStack SEGMENT STACK

DW 12 DUP(?)

AStack ENDS

;Данные программы

DATA SEGMENT

;Директивы описания данных

string DB 15, 15 DUP('$')

sign DB 1

a DW 2

b DW 1

i DW 3

k DW -1

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 f1

mov cx, i

mov ax, cx

shl cx, 1

mov bx, b ;

cmp a, bx ; сравнение a и b

jle f2second

neg cx

add cx, 15

jmp f2final

f2second:

add cx, ax

add cx, 4

f2final:

mov i1, cx

;вычисление f7

mov cx, i

cmp a, bx

jle f4second

shl cx, 1

shl cx, 1

add cx, -5

neg cx

jmp f4final

f4second:

mov ax, cx

shl cx, 1

add cx, ax

neg cx

add cx, 10

f4final:

mov i2, cx

;рассчет f6

mov bx, k

cmp bx, 0

jl f7Second

mov cx, i2

cmp cx, 0

jge skip2 ;модуль i2

neg cx

mov i2, cx

skip2:

mov bx, i2

cmp bx, 7

jl max1

mov cx, bx ; |i2| >= 7

jmp MainFinal

max1:

mov cx, 7 ; |i2| < 7

jmp MainFinal

f7Second:

mov cx, i2

neg cx

add cx, i1

cmp cx, 0

jge skip1

neg cx

skip1:

MainFinal: ; в cx лежит значение функции f7

ret

Main ENDP

CODE ENDS

END Main

Название файла: lab3List.lst

\_\_Microsoft (R) Macro Assembler Version 5.10 11/17/21 16:04:1

Page 1-1

; РЎС‚РµРє РїСЂРѕРіСЂР°РјРјС‹

0000 AStack SEGMENT STACK

0000 000C[ DW 12 DUP(?)

????

]

0018 AStack ENDS

;Р”Р°РЅРЅС‹Рµ РїСЂРѕРіСЂР°РјРјС‹

0000 DATA SEGMENT

;Р”РёСЂРµРєС‚РёРІС‹ РѕРїРёСЃР°РЅРёСЏ РґР°РЅРЅС‹

С…

0000 0F string DB 15, 15 DUP('$')

000F[

24

]

0010 01 sign DB 1

0011 0001 a DW 1

0013 0002 b DW 2

0015 0003 i DW 3

0017 FFFF k DW -1

0019 0000 i1 DW 0

001B 0000 i2 DW 0

001D DATA ENDS

; РљРѕРґ РїСЂРѕРіСЂР°РјРјС‹

0000 CODE SEGMENT

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

; Р“РѕР»РѕРІРЅР°СЏ РїСЂРѕС†РµРґСѓСЂР°

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

;РІС‹С‡РёСЃР»РµРЅРёРµ f1

000C 8B 0E 0015 R mov cx, i

0010 8B C1 mov ax, cx

0012 D1 E1 shl cx, 1

0014 8B 1E 0013 R mov bx, b ;

0018 39 1E 0011 R cmp a, bx ; СЃСЂР°РІРЅРµРЅРёРµ a Р

ё b

001C 7E 08 jle f2second

001E F7 D9 neg cx

0020 83 C1 0F add cx, 15

0023 EB 06 90 jmp f2final

\_\_Microsoft (R) Macro Assembler Version 5.10 11/17/21 16:04:1

Page 1-2

0026 f2second:

0026 03 C8 add cx, ax

0028 83 C1 04 add cx, 4

002B f2final:

002B 89 0E 0019 R mov i1, cx

;РІС‹С‡РёСЃР»РµРЅРёРµ f7

002F 8B 0E 0015 R mov cx, i

0033 39 1E 0011 R cmp a, bx

0037 7E 0C jle f4second

0039 D1 E1 shl cx, 1

003B D1 E1 shl cx, 1

003D 83 C1 FB add cx, -5

0040 F7 D9 neg cx

0042 EB 0C 90 jmp f4final

0045 f4second:

0045 8B C1 mov ax, cx

0047 D1 E1 shl cx, 1

0049 03 C8 add cx, ax

004B F7 D9 neg cx

004D 83 C1 0A add cx, 10

0050 f4final:

0050 89 0E 001B R mov i2, cx

;СЂР°СЃСЃС‡РµС‚ f6

0054 8B 1E 0017 R mov bx, k

0058 83 FB 00 cmp bx, 0

005B 7C 23 jl f7Second

005D 8B 0E 001B R mov cx, i2

0061 83 F9 00 cmp cx, 0

0064 7D 06 jge skip2 ;РјРѕРґСѓР»СЊ i2

0066 F7 D9 neg cx

0068 89 0E 001B R mov i2, cx

006C skip2:

006C 8B 1E 001B R mov bx, i2

0070 83 FB 07 cmp bx, 7

0073 7C 05 jl max1

0075 8B CB mov cx, bx ; |i2| >= 7

0077 EB 18 90 jmp MainFinal

007A max1:

007A B9 0007 mov cx, 7 ; |i2| < 7

007D EB 12 90 jmp MainFinal

0080 f7Second:

0080 8B 0E 001B R mov cx, i2

0084 F7 D9 neg cx

0086 03 0E 0019 R add cx, i1

008A 83 F9 00 cmp cx, 0

008D 7D 02 jge skip1

008F F7 D9 neg cx

0091 skip1:

\_\_Microsoft (R) Macro Assembler Version 5.10 11/17/21 16:04:1

Page 1-3

0091 MainFinal: ; РІ cx Р»РµР¶Рё

С‚ Р·РЅР°С‡РµРЅРёРµ С„СѓРЅРєС†РёРё f7

0091 CB ret

0092 Main ENDP

0092 CODE ENDS

END Main

\_\_Microsoft (R) Macro Assembler Version 5.10 11/17/21 16:04:1

Symbols-1

Segments and Groups:

N a m e Length Align Combine Class

ASTACK . . . . . . . . . . . . . 0018 PARA STACK

CODE . . . . . . . . . . . . . . 0092 PARA NONE

DATA . . . . . . . . . . . . . . 001D PARA NONE

Symbols:

N a m e Type Value Attr

A . . . . . . . . . . . . . . . L WORD 0011 DATA

B . . . . . . . . . . . . . . . L WORD 0013 DATA

F2FINAL . . . . . . . . . . . . L NEAR 002B CODE

F2SECOND . . . . . . . . . . . . L NEAR 0026 CODE

F4FINAL . . . . . . . . . . . . L NEAR 0050 CODE

F4SECOND . . . . . . . . . . . . L NEAR 0045 CODE

F7SECOND . . . . . . . . . . . . L NEAR 0080 CODE

I . . . . . . . . . . . . . . . L WORD 0015 DATA

I1 . . . . . . . . . . . . . . . L WORD 0019 DATA

I2 . . . . . . . . . . . . . . . L WORD 001B DATA

K . . . . . . . . . . . . . . . L WORD 0017 DATA

MAIN . . . . . . . . . . . . . . F PROC 0000 CODE Length = 0092

MAINFINAL . . . . . . . . . . . L NEAR 0091 CODE

MAX1 . . . . . . . . . . . . . . L NEAR 007A CODE

SIGN . . . . . . . . . . . . . . L BYTE 0010 DATA

SKIP1 . . . . . . . . . . . . . L NEAR 0091 CODE

SKIP2 . . . . . . . . . . . . . L NEAR 006C CODE

STRING . . . . . . . . . . . . . L BYTE 0000 DATA

@CPU . . . . . . . . . . . . . . TEXT 0101h

@FILENAME . . . . . . . . . . . TEXT hello3

@VERSION . . . . . . . . . . . . TEXT 510

104 Source Lines

104 Total Lines

26 Symbols

47958 + 461349 Bytes symbol space free

0 Warning Errors

0 Severe Errors