

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

Тарасов К.О.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2021

Цель работы

Изучить представление и обработку целых чисел. Научиться организовывать ветвящиеся процессы на языке Ассемблера

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 .

Ход работы

Перед вычислением функций в sx записывается значение i , после чего значение записанное в sx , с помощью сдвига, умножается на 4 для дальнейших вычислений. При помощи команды `jle` программы переходят к соответствующей метке. Далее идёт подсчёт значений функций $f1$ и $f2$ и запись их в $i1$ и $i2$. Далее по значению k выбирается принимаемые вычисления функции res .

Вариант 25

$f5 = 20 - 4*i$, при $a > b$, $-(6*i - 6)$, при $a \leq b$

$f8 = -(6*i + 8)$, при $a > b$, $9 - 3*(i - 1)$, при $a \leq b$

$f6 = |i1 - i2|$, при $k < 0$, $\max(7, |i2|)$, при $k \geq 0$

Тестирование

Табл. 1. Результат тестирования.

Номер	Входные данные	Результат	Комментарий
-------	----------------	-----------	-------------

1	a = 1 b = 2 i = 3 k = 1	I1 = -12 i2 = 3 res = 7	Верно
2	A = 2 b = 1 I = 3 k = 1	I1 = 8 i2 = -26 res = 26	Верно
3	A = 1 b = 2 I = 2 k = -1	L1 = -6 l2 = 6 res = 12	Верно
4	A = 2 b = 1 I = 4 k = -1	L1 = 4 l2 = -32 res = 36	Верно

Вывод

Изучено представление и обработка целых чисел. В ходе выполнения работы были получены навыки в организации ветвящихся процессов на языке Ассемблера

ПРИЛОЖЕНИЕ А

Текст компонентов программы lr3.exe

lr3.asm

; Стек программы

AStack SEGMENT STACK

DW 12 DUP(?)

AStack ENDS

;Данные программы

DATA SEGMENT

;Директивы описания данных

a DW 2

b DW 1

i DW 4

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, i
shl cx, 1
shl cx, 1
mov bx, b
cmp a, bx
;f5 = 20 - 4*i , при a>b, -(6*I - 6), при a<=b
;f8 = - (6*i+8) , при a>b, 9 -3*(i-1), при a<=b
jle f85ch
    mov ax, cx
neg cx
    add cx, 20
    mov i1, cx
    add ax, i
    add ax, i
    add ax, 8
    neg ax
    mov i2, ax

    jmp f85chf
f85ch:
    mov ax, i
    mov i2, ax
    add i2, -1
    shl i2, 1
    add i2, ax
    add i2, 9

    add cx, i
    add cx, i
    add cx, -6

```

```

        neg cx
        mov i1, cx
f85chf:

;f6 = |i1 - i2|, при k<0 , max(7, |i2|), при k>=0
mov bx, k
cmp bx, 0
jl f6ch
    mov bx, i2
    cmp bx, 0
    jge temp1
    neg bx
temp1:
    cmp bx, 7
    jl max1
        mov cx, bx
        jmp f6chf
max1:
    mov cx, 7
    jmp f6chf
f6ch:
    mov cx, i1
    sub cx, i2
    cmp cx, 0
    jge temp2
    neg cx
temp2:
f6chf:
ret
Main    ENDP

```

CODE ENDS

END Main

CODE ENDS

END Main

lr3.lst:

#Microsoft (R) Macro Assembler Version 5.10

12/16/21 03:14:4

Page 1-1

; Стек программы

0000 AStack SEGMENT STACK

0000 000C[DW 12 DUP(?)

????

]

0018 AStack ENDS

;Данные программы

0000 DATA SEGMENT

;Директивы описания данны

x

0000 0002 a DW 2

0002 0001 b DW 1

0004 0004 i DW 4

0006 FFFF k DW -1

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

                                ; Главная 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 8B 0E 0004 R                      mov  cx, i
000D D1 E1                                shl  cx, 1
000F D1 E1                                shl  cx, 1
0011 8B 1E 0002 R                      mov  bx, b
0015 39 1E 0000 R                      cmp  a, bx

                                ;f5 = 20 - 4*i , при a>b, -(6*I - 6)
                                , при a<=b

                                ;f8 = - (6*i+8) , при a>b, 9 -3*(i-1
                                ), при a<=b

0019 7E 1E                                jle  f85ch
001B 8B C1                                mov  ax, cx
001D F7 D9                                neg  cx
001F 83 C1 14                            add  cx, 20
0022 89 0E 0008 R                      mov  i1, cx
0026 03 06 0004 R                      add  ax, i
002A 03 06 0004 R                      add  ax, i
002E 05 0008                            add  ax, 8
0031 F7 D8                                neg  ax
0033 A3 000A R                        mov  i2, ax

```


0036	EB 2A 90	jmp f85chf
0039		f85ch:
0039	A1 0004 R	mov ax, i

```
003C A3 000A R          mov i2, ax
003F 83 06 000A R FF      add i2, -1
0044 D1 26 000A R          shl i2, 1
0048 01 06 000A R          add i2, ax
004C 83 06 000A R 09      add i2, 9
```

```
0051 03 0E 0004 R          add cx, i
0055 03 0E 0004 R          add cx, i
0059 83 C1 FA            add cx, -6
005C F7 D9              neg cx
005E 89 0E 0008 R          mov i1, cx
```

```
0062                  f85chf:
```

;f6 = |i1 - i2|, при k<0 , max(7, |i
2|), при k>=0

```
0062 8B 1E 0006 R          mov bx, k
0066 83 FB 00            cmp bx, 0
0069 7C 1B              jl f6ch
006B 8B 1E 000A R          mov bx, i2
006F 83 FB 00            cmp bx, 0
0072 7D 02              jge temp1
0074 F7 DB              neg bx
0076                  temp1:
0076 83 FB 07            cmp bx, 7
0079 7C 05              jl max1
007B 8B CB              mov cx, bx
```

```

007D EB 16 90                                jmp f6chf
0080                                max1:
0080 B9 0007                                mov cx, 7
0083 EB 10 90                                jmp f6chf
0086                                f6ch:
0086 8B 0E 0008 R                            mov cx, i1
008A 2B 0E 000A R                            sub cx, i2
008E 83 F9 00                                cmp cx, 0
0091 7D 02                                jge temp2
0093 F7 D9                                neg cx
0095                                temp2:
0095                                f6chf:
0095 CB                                ret
0096                                Main    ENDP
0096                                CODE    ENDS
                                END Main

```

Symbols-1

Segments and Groups:

N a m e	Length	Align	Combine	Class
ASTACK	0018	PARA		STACK
CODE	0096	PARA		NONE
DATA	000C	PARA		NONE

Symbols:

N a m e	Type	Value	Attr
A	L WORD	0000	DATA
B	L WORD	0002	DATA
F6CH	L NEAR	0086	CODE
F6CHF	L NEAR	0095	CODE
F85CH	L NEAR	0039	CODE
F85CHF	L NEAR	0062	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 = 0096
MAX1	L NEAR	0080	CODE	
TEMP1	L NEAR	0076	CODE	
TEMP2	L NEAR	0095	CODE	
@CPU	TEXT	0101h		
@FILENAME	TEXT	1r3		
@VERSION	TEXT	510		

89 Source Lines

89 Total Lines

22 Symbols

47990 + 461317 Bytes symbol space free

0 Warning Errors

0 Severe Errors