

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, 0
```

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

```
mov cx, i
```

```
mov ax, cx
```

```
shl cx, 1
```

```
shl cx, 1
```

```
mov bx, b
```

```
cmp a, bx
```

```
jle f5ch
```

```
neg cx
```

```
add cx, 20
```

```
jmp f5chf
```

```
f5ch:
```

```
add cx, ax
```

```
add cx, ax
```

```
add cx, -6
```

```
neg cx
```

```
f5chf:
```

```
mov i1, cx
```

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

```
mov cx, i
```

```
cmp a, bx
```

```
jle f8ch
```

```
shl cx, 1
```

```
mov ax, cx
```

```
shl cx, 1
```

```
add cx, ax
```

```

        add cx, 8
        neg cx
        jmp f8chf
f8ch:
        add cx, -1
        mov ax, cx
        shl cx, 1
        add cx, ax
        neg cx
        add cx, 9
f8chf:
        mov i2, cx

;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 cx
temp1:
        cmp bx, 7
        jl max1
        mov cx, bx
        jmp f6chf
max1:
        mov cx, 7
        jmp f6chf
f6ch:

```

```

        mov cx, i2
        neg cx
        add cx, i1
        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

11/6/21 08:21:46

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
```


; 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
```

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

```
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
001E 7E 08          jle  f5ch
0020 F7 D9          neg  cx
0022 83 C1 14       add  cx, 20
0025 EB 0A 90       jmp  f5chf
0028              f5ch:
0028 03 C8          add  cx, ax
002A 03 C8          add  cx, ax
002C 83 C1 FA       add  cx, -6
002F F7 D9          neg  cx
0031              f5chf:
0031 89 0E 0008 R   mov  i1, cx
```

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

```
0035 8B 0E 0004 R      mov cx, i
0039 39 1E 0000 R      cmp a, bx
003D 7E 10             jle f8ch
003F D1 E1             shl cx, 1
0041 8B C1             mov ax, cx
0043 D1 E1             shl cx, 1
0045 03 C8             add cx, ax
0047 83 C1 08          add cx, 8
004A F7 D9            neg cx
004C EB 0F 90          jmp f8chf
004F                  f8ch:
004F 83 C1 FF          add cx, -1
0052 8B C1             mov ax, cx
0054 D1 E1             shl cx, 1
0056 03 C8             add cx, ax
0058 F7 D9            neg cx
005A 83 C1 09          add cx, 9
005D                  f8chf:
005D 89 0E 000A R      mov i2, cx
```

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

```
0061 8B 1E 0006 R      mov bx, k
```

0065	83 FB 00	cmp bx, 0
0068	7C 1B	jl f6ch
006A	8B 1E 000A R	mov bx, i2
006E	83 FB 00	cmp bx, 0
0071	7D 02	jge temp1
0073	F7 D9	neg cx
0075		temp1:
0075	83 FB 07	cmp bx, 7
0078	7C 05	jl max1
007A	8B CB	mov cx, bx
007C	EB 18 90	jmp f6chf
007F		max1:
007F	B9 0007	mov cx, 7
0082	EB 12 90	jmp f6chf
0085		f6ch:
0085	8B 0E 000A R	mov cx, i2
0089	F7 D9	neg cx
008B	03 0E 0008 R	add cx, i1
008F	83 F9 00	cmp cx, 0
0092	7D 02	jge temp2
0094	F7 D9	neg cx
0096		temp2:
0096		f6chf:
0096	CB	ret
0097		Main ENDP
0097		CODE ENDS
		END Main

Symbols-1

Segments and Groups:

N a m e	Length	Align	Combine	Class
ASTACK	0018	PARA		STACK
CODE	0097	PARA		NONE
DATA	000C	PARA		NONE

Symbols:

N a m e	Type	Value	Attr
A	L WORD	0000	DATA
B	L WORD	0002	DATA
F5CH	L NEAR	0028	CODE
F5CHF	L NEAR	0031	CODE
F6CH	L NEAR	0085	CODE
F6CHF	L NEAR	0096	CODE
F8CH	L NEAR	004F	CODE
F8CHF	L NEAR	005D	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 = 0097

MAX1 L NEAR 007F CODE

TEMP1 L NEAR 0075 CODE

TEMP2 L NEAR 0096 CODE

@CPU TEXT 0101h

@FILENAME TEXT 1r3

@VERSION TEXT 510

99 Source Lines

99 Total Lines

24 Symbols

47990 + 461317 Bytes symbol space free

0 Warning Errors

0 Severe Errors