

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 19

Студентка гр. 1383

Седова Э.А.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

Цель работы.

Изучение понятия ветвящихся процессов. Разработка программы, обрабатывающей целые числа, 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 .

Выполнение работы.

$$I1 = f4 = \begin{cases} / -(6*i - 4), & \text{при } a > b \\ \backslash 3*(i+2), & \text{при } a \leq b \end{cases}$$

$$I2 = f5 = \begin{cases} / 20 - 4*i, & \text{при } a > b \\ \backslash -(6*I - 6), & \text{при } a \leq b \end{cases}$$

$$Res = f7 = \begin{cases} / |i1| + |i2|, & \text{при } k < 0 \\ \backslash \max(6, |i1|), & \text{при } k \geq 0 \end{cases}$$

Таблица 1 – примеры тестовых случаев

Номер	Входные данные	Выходные данные
1	$a=5$ $b=8$ $i=2$	$i1=000C=12$ $i2=FFFA=-6$ $res = 0034=12$

	k=3	
2	a=2 b=-3 i=-8 k=-1	i1=0034=52 i2=0034=52 res = 0034=104
3	a=-40 b=0 i=1 k=1	i1=0009=9 i2=0000=0 res = 0009=9
4	a=0 b=0 i=0 k=0	i1=0006=6 i2=0006=6 res = 0006=6

Для минимализации длины кода были произведены следующие упрощения:

1. При $a > b$:

$$-(6i - 4) = -((2i + i)2 - 4)$$

$$20 - 4i = -4(i - 5)$$

2. При $a \leq b$:

$$\text{Пусть } 3(i+2) = x, \text{ тогда } -(6i - 6) = 2(9 - x)$$

3. Так как в f3 всегда используется модуль f1, метка getabs_i1, необходимая для взятия модуля от f1, вынесена отдельно. Это помогает избежать дублирования кода.

Программный код см. в приложении А.

Файлы диагностических сообщений см. в приложении Б.

Выводы.

В ходе выполнения работы было изучено понятие ветвящихся процессов.
Разработана программа, на языке Ассемблера, обрабатывающая целые числа.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: 3lb.asm

```

DOSSEG                                     ; Задание сегментов под
ДОС
        .MODEL    SMALL                     ; Модель памяти-
SMALL (Малая)
        .STACK    100h                     ; Отвести под Стек
256 байт
        .DATA     ; Начало сегмента
данных
        a dw 0
        b dw 0
        i dw 0
        k dw 0
        i1 dw 0
        i2 dw 0
        res dw 0
        .CODE     ; Начало сегмента кода
        mov ax, @data ; Загрузка в DS адреса
начала
        mov ds, ax ; сегмента данных

        mov ax, a
        mov cx, i ;i
        shl cx, 1 ;2i
        add cx, i ;3i
        cmp ax, b ;сравниваем равны ли a и b.
        jle second ;выполняет короткий переход, если первый операнд >
второго операнда
        first:    ;if(a>b)
        shl cx, 1 ;6i
        sub cx, 4 ;6i-4
        neg cx

        mov ax, i ;i
        sub ax, 5 ;i-5
        shl ax, 1 ;2i-10
        shl ax, 1 ;4i-20
        neg ax ;20-4i
        jmp result

        second:   ;if(a<=b)
        add cx, 6 ;3i+6

        mov ax, cx ;3i+6
        neg ax ;-3i-6
        add ax, 9 ;-3i+3
        shl ax, 1 ;-6i+6

        result:
        mov [i1], cx
        mov [i2], ax

```

```

getabs_i1:
    neg cx
    js getabs_i1
    cmp k, 0
jge final2 ;короткий переход, если первый операнд >= второго
операнда
final1:
    getabs_i2:
        neg ax
        js getabs_i2
    add cx, ax
    jmp answer

final2:
    cmp cx, 6
    jge answer
    mov cx, 6
answer:
    mov [res], cx
    mov ah, 4ch
    int 21h
; завершение программы и
ВЫХОД В ДОС
END

```

ПРИЛОЖЕНИЕ Б

ФАЙЛЫ ДИАГНОСТИЧЕСКИХ СООБЩЕНИЙ

Название файла: 3lb.lst

Microsoft (R) Macro Assembler Version 5.10

11/6/22 19:44:31

Page

1-1

```

DOSSEG
P-P°PrP°PSPëPµ CÍPµPiPjPµPSC,PsPI PiPsPr
P°PhPŸ

.MODEL SMALL
; PµPsPrPµP»Cµ PiP°PjCЦC,Pë-SMALL (PµP°P»P°CЦ)
.STACK 100h
; PhC,PIPµCÍC,Pë PiPsPr PŸC,PµPë 256 P±P°PN°C,
.DATA
; PќP°C±P°P»Ps CÍPµPiPjPµPSC,P° PrP°PSPSC<C...

0000 0000 a dw 0
0002 0000 b dw 0
0004 0000 i dw 0
0006 0000 k dw 0
0008 0000 i1 dw 0
000A 0000 i2 dw 0
000C 0000 res dw 0

.CODE
PќP°C

P°P»Ps CÍPµPiPjPµPSC,P° PëPsPrP°
0000 B8 ---- R mov ax, @data ; P-P°
PiCБCÍP·PëP° PI DS P°PrCБPµCÍP° PSP°C±P°P»P°
0003 8E D8 mov ds,
ax ; CÍPµ
PiPjPµPSC,P° PrP°PSPSC<C...

0005 A1 0000 R mov ax, a
0008 8B 0E 0004 R mov cx, i ;i

```

```

000C D1 E1          shl cx, 1 ;2i
000E 03 0E 0004 R   add cx, i ;3i
0012 3B 06 0002 R   cmp ax, b ;CfCbP°PIPSpëPIP°PµPj
CbP°PIpsc

< P»Pë a Pë b.

0016 7E 16          jle second ;PIC<PiPsP»PscµPµC,
PePsCbPsC,P

ePëPN° PiPµCbPµC...PsPr, PµCfP»Pë PiPµCbPIC<PN°
PsP

iPµCbP°PSPr > PIC,PsCbPsPiPs PsPiPµCbP°PSPrP°
0018 first:          ;if(a>b)
0018 D1 E1          shl cx, 1 ;6i
001A 83 E9 04        sub cx, 4 ;6i-4
001D F7 D9          neg cx

001F A1 0004 R       mov ax, i ;i
0022 2D 0005         sub ax, 5 ;i-5
0025 D1 E0          shl ax, 1 ;2i-10
0027 D1 E0          shl ax, 1 ;4i-20
0029 F7 D8          neg ax ;20-4i
002B EB 0D 90        jmp result

002E second:         ;if(a<=b)
002E 83 C1 06        add cx, 6 ;3i+6

0031 8B C1          mov ax, cx ;3i+6
0033 F7 D8          neg ax ;-3i-6
0035 05 0009        add ax, 9 ;-3i+3
0038 D1 E0          shl ax, 1 ;-6i+6

003A result:

```


1-2

```

003A 89 0E 0008 R      mov [i1], cx
003E A3 000A R      mov [i2], ax

0041                  getabs_i1:
0041 F7 D9              neg cx
0043 78 FC              js getabs_i1

                        ;mov bx, k

0045 83 3E 0006 R 00    cmp k, 0
004A 7D 09              jge final2 ;PePsCtPsC,PePëPN
PiPuCtPuC...PsP

                        r, PuCtP»Pë PiPuCtPIC<PN PsPiPuCtP°PSPr >=
PIC,

                        PsCtPsPiPs PsPiPuCtP°PSPrP°

004C                  final1:
004C                  getabs_i2:
004C F7 D8              neg ax
004E 78 FC              js getabs_i2
0050 03 C8              add cx, ax
0052 EB 09 90          jmp answer

0055                  final2:
0055 83 F9 06            cmp cx, 6
0058 7D 03              jge answer
005A B9 0006            mov cx, 6
005D                  answer:
005D 89 0E 000C R      mov [res], cx
0061 B4 4C              mov ah, 4ch
0063 CD 21              int 21h ;

P·P°

                        PIPuCtCëPuPSPëPu PiCtPsPiCtP°PjPjC< Pë
PIC<C...Ps

                        Pr PI P"PtPŸ

```

END

Microsoft (R) Macro Assembler Version 5.10
11/6/22 19:44:31

Symbols-1

Segments and Groups:

N a m e	Length	Align	Combine Class
DGROUP	GROUP		
_DATA	000E WORD	PUBLIC	'DATA'
STACK	0100 PARA	STACK	'STACK'
_TEXT	0065 WORD	PUBLIC	'CODE'

Symbols:

N a m e	Type	Value	Attr
A	L WORD	0000	_DATA
ANSWER	L NEAR	005D	_TEXT
B	L WORD	0002	_DATA
FINAL1	L NEAR	004C	_TEXT
FINAL2	L NEAR	0055	_TEXT
FIRST	L NEAR	0018	_TEXT
GETABS_I1	L NEAR	0041	_TEXT
GETABS_I2	L NEAR	004C	_TEXT
I	L WORD	0004	_DATA
I1	L WORD	0008	_DATA
I2	L WORD	000A	_DATA
K	L WORD	0006	_DATA
RES	L WORD	000C	_DATA

RESULT	L NEAR	003A _TEXT
SECOND	L NEAR	002E _TEXT
@CODE	TEXT	_TEXT
@CODESIZE	TEXT	0
@CPU	TEXT	0101h
@DATASIZE	TEXT	0
@FILENAME	TEXT	_31b
@VERSION	TEXT	510

Microsoft (R) Macro Assembler Version 5.10
11/6/22 19:44:31

Symbols-2

70 Source Lines

70 Total Lines

32 Symbols

48004 + 457206 Bytes symbol space free

0 Warning Errors

0 Severe Errors