

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Алтайский государственный технический университет им. И. И. Ползунова»

Факультет информационных технологий
Кафедра прикладной математики

Отчет защищен с оценкой _____

Преподаватель _____
(подпись)

« ____ » _____ 2022 г.

Отчет
По Лабораторной работе №2

«Разработка и отладка программ на ассемблере»

по дисциплине «Машинно-зависимые языки программирования - 4 семестр»

Студент группы ПИ-02 Чередов Р.А.

Преподаватель доцент, к.т.н. Боровцов Е.Г.

Барнаул 2022

Задача:

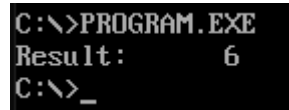
Реализовать линейный, ветвящийся и циклический алгоритмы на ассемблере в базовом наборе команд для 16-разрядной среды DOS

Для ввода и вывода исходных данных и результатов использовать функции 0Ah и 09h прерывания DOS Int 21h

Для конвертации текстовых строк в число и числа в текстовую строку использовать подпрограммы str2bin и bin2str.

Линейный алгоритм:

Заданное выражение: $Y = a * b + (c^2 - d)$



Данные: $A = 2, B = 2, C = 2, D = 2$

Код программы:

```
.model    small                ; Designate the application as EXE file

,
, *****
; Our stack section
, *****
.stack

,
, *****
; Our data section. Here we declare our strings for our console message
, *****
.data
messin_A    db 'Enter Number A: $'      ; Сообщение при вводе A
messin_B    db 'Enter Number B: $'      ; Сообщение при вводе B
messin_C    db 'Enter Number C: $'      ; Сообщение при вводе C
messin_D    db 'Enter Number D: $'      ; Сообщение при вводе D
messout     db 'Result: $'              ; Сообщение при выводе результата
           db 256 dup(?)                ; Выделение памяти для стека
numb_A      dw ?                        ; Переменная для хранения числа которое вводит пользователь
numb_B      dw ?                        ; Переменная для хранения числа которое вводит пользователь
numb_C      dw ?                        ; Переменная для хранения числа которое вводит пользователь
numb_D      dw ?                        ; Переменная для хранения числа которое вводит пользователь
result      db 10 dup(?),'$'            ; Переменная для вывода результата, здесь лежит число после
преобразования в строку
input_Buf   db 06,00,5 dup(?)           ; Переменная для хранения числа пользователя

.code

include bin2str.asm ; подпрограмма преобразования числа 2сфт в строку
include str2bin.asm ; подпрограмма преобразования строки в двоичное число
main        proc
; Ввод данных (до строки с "*****...")
; Выделения памяти для стека
mov ax,@data
mov ds,ax
; Вывод сообщения которое находится в переменной messin_A
lea dx, offset messin_A
mov ah, 09
int 21h
; Вызов команды ввода
lea dx,input_Buf
mov ah, 0ah
int 21h
; Преобразование введенной строки в число
mov bx,dx
inc bx
```

```

call str2bin
mov numb_A,ax ; Результат преобразования сохраняется по метке

; Вывод сообщения которое находится в переменной messin_B
lea dx, offset messin_B
mov ah, 09
int 21h
; Вызов команды ввода
lea dx,input_Buf
mov ah, 0ah
int 21h
; Преобразование введённой строки в число
mov bx,dx
inc bx
call str2bin
mov numb_B,ax ; Результат преобразования сохраняется по метке
; Вывод сообщения которое находится в переменной messin_C
lea dx, offset messin_C
mov ah, 09
int 21h
; Вызов команды ввода
lea dx,input_Buf
mov ah, 0ah
int 21h
; Преобразование введённой строки в число
mov bx,dx
inc bx
call str2bin
mov numb_C,ax ; Результат преобразования сохраняется по метке
; Вывод сообщения которое находится в переменной messin_D
lea dx, offset messin_D
mov ah, 09
int 21h
; Вызов команды ввода
lea dx,input_Buf
mov ah, 0ah
int 21h
; Преобразование введённой строки в число
mov bx,dx
inc bx
call str2bin
mov numb_D,ax ; Результат преобразования сохраняется по метке
, *****
; Вычисления (До строки с "***** ...")
mov ax,numb_C ;ax -> C
mul numb_C ;c*c
mov bx,numb_D ; bx -> D
sub ax,bx ;C^2-D
mov bx,ax ; ax -> bx
mov ax,numb_A ; numb_A -> ax
mul numb_B ; ax*numb_B
add ax,bx ; ax + bx
, *****
; Вывод результатов (до конца программы)
; Преобразование числа которое находится в ax в строку для вывода в консоль
lea bx,result
call bin2str
; Вывод сообщения которое находится в переменной messout
lea dx, offset messout
mov ah, 09
int 21h
; Вывод числа
lea dx,result+1
mov ah,09
int 21h

```

```

; Выход из программы
mov     ax,4c00h
int     21h

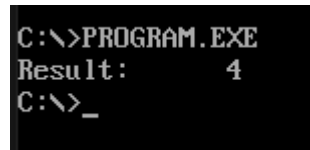
main     endp
end       main

```

Ветвящийся алгоритм:

Заданное выражение:

$$Y: = \begin{cases} a*b*c/d, & \text{если } C \neq 0 \text{ и } D \neq 0 \\ A*B, & \text{если } C=0 \text{ и } D=0 \end{cases}$$



$A = 2, B = 2, C = 0, D = 2$



$A = 2, B = 2, C = 3, D = 2$

Код программы:

```
.model small
```

```
.stack
```

```
, ****
```

```
; Our data section. Here we declare our strings for our console message
```

```
, ****
```

```
.data
```

```
messin_A      db 'Enter Number A: $'      ; Сообщение при вводе A
```

```
messin_B      db 'Enter Number B: $'      ; Сообщение при вводе B
```

```
messin_C      db 'Enter Number C: $'      ; Сообщение при вводе C
```

```
messin_D      db 'Enter Number D: $'      ; Сообщение при вводе D
```

```
messout db 'Result: $'      ; Сообщение при выводе результата
```

```
db 256 dup(?) ; Выделение памяти для стека
```

```
numb_A        dw ?          ; Переменная для хранения числа которое вводит пользователь
```

```
numb_B        dw ?          ; Переменная для хранения числа которое вводит пользователь
```

```
numb_C        dw ?          ; Переменная для хранения числа которое вводит пользователь
```

```
numb_D        dw ?          ; Переменная для хранения числа которое вводит пользователь
```

```
result        db 10 dup(?),'$' ; Переменная для вывода результата, здесь лежит число после преобразования в строку
```

```
input_Buf      db 06,00,5 dup(?)      ; Переменная для хранения числа пользователя
```

```
.code
```

```
include bin2str.asm ; подпрограмма преобразования числа 2сфт в строку
```

```
include str2bin.asm ; подпрограмма преобразования строки в двоичное число
```

```
main      proc
```

```
; Ввод данных (до строки с "***** ...")
```

```
        ; Выделения памяти для стека
```

```
        mov ax,@data
```

```
        mov ds,ax
```

```
        ; Вывод сообщения которое находится в переменной messin_A
```

```
        lea dx, offset messin_A
```

```
        mov ah, 09
```

```
        int 21h
```

```
        ; Вызов команды ввода
```

```
        lea dx,input_Buf
```

```
        mov ah, 0ah
```

```
        int 21h
```

```
        ; Преобразование введённой строки в число
```

```
        mov bx,dx
```

```
        inc bx
```

```
        call str2bin
```

```
        mov numb_A,ax ; Результат преобразования сохраняется по метке
```

```
        ; Вывод сообщения которое находится в переменной messin_B
```

```
        lea dx, offset messin_B
```

```
        mov ah, 09
```

```
        int 21h
```

```
        ; Вызов команды ввода
```

```
        lea dx,input_Buf
```

```
mov ah, 0ah

int 21h

; Преобразование введённой строки в число

mov bx,dx

inc bx

call str2bin

mov numb_V,ax ; Результат преобразования сохраняется по метке

; Вывод сообщения которое находится в переменной messin_C

lea dx, offset messin_C

mov ah, 09

int 21h

; Вызов команды ввода

lea dx,input_Buf

mov ah, 0ah

int 21h

; Преобразование введённой строки в число

mov bx,dx

inc bx

call str2bin

mov numb_C,ax ; Результат преобразования сохраняется по метке

; Вывод сообщения которое находится в переменной messin_D

lea dx, offset messin_D

mov ah, 09

int 21h

; Вызов команды ввода

lea dx,input_Buf

mov ah, 0ah

int 21h

; Преобразование введённой строки в число
```

```

mov bx,dx

inc bx

call str2bin

mov numb_D,ax ; Результат преобразования сохраняется по метке

, *****

; Вычисления (До строки с "***** ...")

mov ax, numb_A ; numb_A -> ax

imul     numb_B      ; ax * numb_B

mov bx,ax; bx->ax

mov ax,numb_C ; numb_C -> ax

imul numb_D ; ax * numb_D

cmp ax, 0 ;ax=0?

jz out_ ; переход по метке out_ если C*D=0

mov ax,bx ;ax->bx

imul numb_C ; ax*C

idiv numb_D ;ax/D

mov bx,ax ;bx->ax

, *****

; Вывод результатов (до конца программы)

; Преобразование числа которое находится в ax в строку для вывода в консоль

out_: mov ax,bx;ax->bx

      lea bx,result

      call bin2str

; Вывод сообщения которое находится в переменной messout

lea dx, offset messout

mov ah, 09

int 21h

; Вывод числа

lea dx,result+1

```

```

mov ah,09

int 21h

; Выход из программы

mov     ax,4c00h

int     21h

main    endp

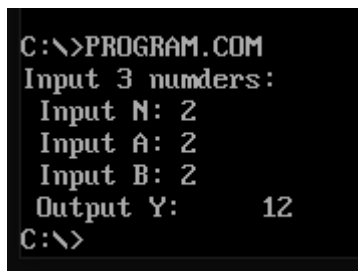
end     main

```

Циклический алгоритм:

$$Y = \sum_{i=1}^N (A*i + B*i)$$

Заданное выражение:



```

C:\>PROGRAM.COM
Input 3 numders:
Input N: 2
Input A: 2
Input B: 2
Output Y:      12
C:\>

```

$A = 2, B = 2, N = 2$

Код программы:

```

.model tiny                ; MASM's support for COM files
.dosseg                   ; Make sure you are using dos segment CODE, DATA + STACK

; *****
; Our data section. Here we declare our strings for our console message
; *****
.data
    strMessage    DB "Input 3 numders: ", 0Dh, 0Ah, '$'
    InpA          db " Input A: ", '$'
    InpB          db " Input B: ", '$'
    InpN          db " Input N: ", '$'
    InpD          db " Input D: ", '$'
    Output        db " Output Y: ", '$'
    NewStr        db " ", 0Dh, 0Ah, '$'
    VarA          dw 0
    VarB          dw 0
    VarN          dw 0
    Res           dw 0
    Buffer         dw 06,0,7 dup(' '), '$'

; *****
; Our executable assembly code starts here in the .code section
; *****
.code

.startup
; *****
; Prepare our message to the console
; *****
    mov     ax, 09h                ; Write string to standard output
    mov     dx, offset strMessage
    int     21h                   ; Call DOS interrupt 21h

```



```

; *****
; Input N
mov             ah, 09h                ; Write string to standard output
mov             dx, offset InpN
int             21h                    ; Call DOS interrupt 21h
mov             ah, 0ah
mov             dx, Buffer
int             21h
mov             bx, dx
inc             bx
call            str2bin
mov             word ptr VarN, ax
mov             ah, 09h                ; Write string to standard output
mov             dx, offset NewStr
int             21h                    ; Call DOS interrupt 21h

; Input A
mov             ah, 09h                ; Write string to standard output
mov             dx, offset InpA
int             21h                    ; Call DOS interrupt 21h
mov             ah, 0ah
mov             dx, Buffer
int             21h
mov             bx, dx
inc             bx
call            str2bin
mov             word ptr VarA, ax
mov             ah, 09h                ; Write string to standard output
mov             dx, offset NewStr
int             21h                    ; Call DOS interrupt 21h

; Input B
mov             ah, 09h                ; Write string to standard output
mov             dx, offset InpB
int             21h                    ; Call DOS interrupt 21h
mov             ah, 0ah
mov             dx, Buffer
int             21h
mov             bx, dx
inc             bx
call            str2bin
mov             word ptr VarB, ax
mov             ah, 09h                ; Write string to standard output
mov             dx, offset NewStr
int             21h                    ; Call DOS interrupt 21h

mov             cx, VarN                ; копировать ячейку (N) в CX (регистр для счетчика цикла)
mov             bx, 0                  ; поместить 0 в BX
mov             Res, bx                ; копировать BX (0) в ячейку (B) (ПОД ВЫХОДНУЮ СУММУ)
cmp             cx, 1                  ; сравнить AX (N) с 1
jl             ext                     ; переход на вывод результата по не положительному результату
cycl: mov         ax, VarA ;           ; начало цикла, копировать ячейку (A) (A) в AX
mov             bx, cx                 ; копировать CX в BX (i)
cmp             bx, 0                  ; сравнить (i) с нулем
jz             ext                     ; переход по нулю на вывод результата
imul            bx                     ; умножить AX на BX ((A) * i)
add             Res, ax                 ; сложить AX с ячейкой (Res) (НАКАПЛИВАЕМАЯ СУММА)
mov             ax, VarB                ; копировать ячейку (B) (B) в AX
mov             bx, cx                 ; копировать CX в BX (i)
imul            bx                     ; умножить AX на BX ((B) * i)
add             ax, Res                 ; сложить AX с ячейкой (Res) (НАКАПЛИВАЕМАЯ СУММА)
mov             res, ax                 ; копировать AX в ячейку (Res) (НАКАПЛИВАЕМАЯ СУММА)
loop            cycl                   ; замкнуть цикл

```

```

ext: mov ah, 09h                                ; Write string to standard output
mov dx, offset Output
int 21h                                           ; Call DOS interrupt 21h
mov ax, Res
lea bx, Buffer
call bin2str
lea dx, Buffer+1
mov ah, 09
int 21h
;
; *****
; Get ready to exit and return control back to DOS
; *****
mov ah, 4Ch                                     ; Exit with return code (AL = return code)
int 21h                                           ; Call DOS interrupt 21h
include str2bin.asm
include bin2str.asm
end

```