

Защищено:

Демонстрация ЛР:

Большаков С.А.

Большаков С.А.

13 марта 2023 г.

**Отчет по лабораторной работе № 3 по курсу  
Системное программирование**

**" Вывод трех символов "**

**(есть ли дополнительные требования- ДА/НЕТ)**

8  
(количество листов)  
Вариант № <10>

ИСПОЛНИТЕЛЬ:

студент группы **ИУ5-41Б**

**Никулин Д.Д.**

\_\_\_\_\_  
(подпись)

"03" \_\_\_\_ апреля \_\_\_\_ 2023 г.

## СОДЕРЖАНИЕ

1. Цель выполнения лабораторной работы № 3.....	3
2. Порядок и условия проведения работы № 3 .....	3
3. Описание ошибок, возникших при отладке № 3 .....	3
4. Блок-схема программы.....	4
5. Скриншот программы в TD.exe.....	6
6. Текст программы на языке Ассемблера .....	6
7. Результаты работы программы.....	8
8. Выводы по ЛР № 3.....	8

## 1.Цель выполнения лабораторной работы № 3

Целью работы является разработка и отладка программы на языке Ассемблер, и изучение способа написания текста программы, выполнения ее компиляции, редактирования связей и компоновки исполняемого модуля программы.

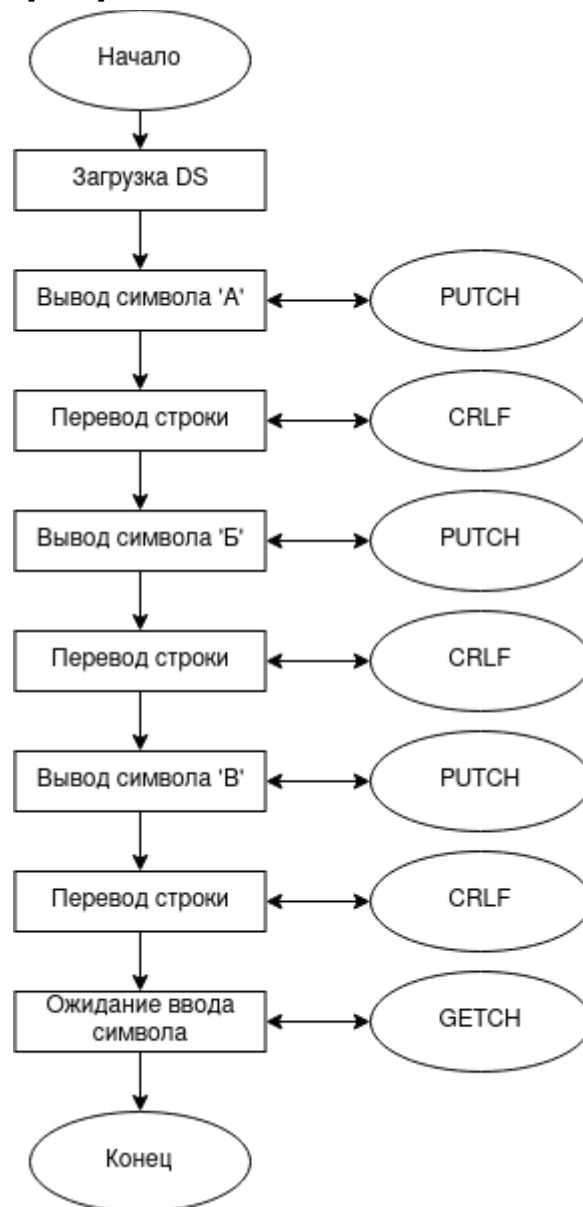
## 2. Порядок и условия проведения работы № 3

Необходимо использовать процедуры при разработке программы. Предусмотреть минимально три процедуры: для ввода символа (1-я процедура - GETCH название процедуры ввода символа желательно взять такое название), для вывода одного символа (2-я процедура - PUTCH) и для перевода строк и с возвратом каретки (3-я процедура - CLRF) на дисплее(оформление процедур - PROC - ENDP, вызов процедур - CALL). После вывода букв программа переходит в состояние ожидания нажатия любой клавиши. Выход из программы осуществить посредством системного прерывания 21H - 04CH с указанием кода возврата 0.Ввод символа используется для организации ожидания завершения программы и просмотра результата.

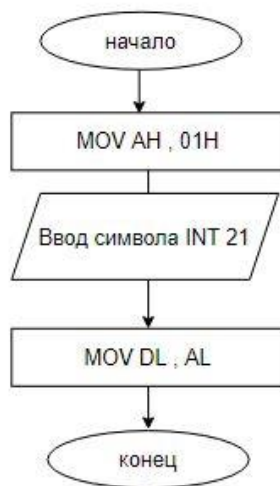
## 3. Описание ошибок, возникших при отладке № 3

№п/п	Проявление ошибки	Причина ошибки	Способ устранения
1.	<b>**error**</b> illegal instruction	Не корректная запись команды	Исправление записи команды
2.	Error: undefined symbol ... in module	Доступ ко внешней переменной, ссылка на которую не была найдена	Ввод в программе корректного сегментного регистра через ASSUME CS:... DS:... и PUSH CS, POP DS

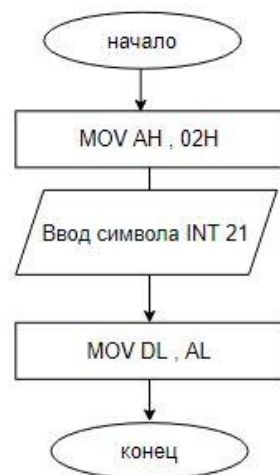
## 4.Блок-схема программы



## GETCH



## PUTCH



## CLRF



## 5.Скриншот программы в TD.exe

File Edit View Run Breakpoints Data Options Window Help		CPU 80486		3
[F5]=Module: firstd File: 8=1=[F7][F8]		#firstd#start: PUSH		ax 0000 c=0
аботу выполнил Никулин ИУ5-41Б Лабор		cs:0001 push cs		bx 0000 z=0
ODE SEGMENT 'CODE'		#firstd#9: POP DS		cx 0000 s=0
ASSUME CS:MYCODE, DS:MYCODE		cs:0002 pop ds		dx 0000 o=0
DB 'N'		#firstd#11: CALL PRI		si 0000 p=0
RT:		cs:0003 call #fir		di 0000 a=0
агрузка сегментного регистра данных		#firstd#12: CALL CRL		bp 0000 i=1
PUSH CS		cs:0006 call #fir		sp 0000 d=0
POP DS		#firstd#13: ADD LET,		ds 4EDD
ывод одного символа на экран		cs:0009 add byte		es 4EDD
CALL PRINTLET		#firstd#15: CALL PRI		ss 4EEC
CALL CRLF		cs:000E call #fir		cs 4EED
ADD LET, 1		#firstd#16: CALL CRL		ip 0001
ывод второго символа на экран		ds:0000 CD 20 FF 9F		
CALL PRINTLET		ds:0008 AD DE E0 01		
CALL CRLF		ds:0010 C5 15 89 02		ss:0002 6474
ADD LET, 1		ds:0018 01 01 01 00		ss:0000 0000
ывод третьего символа на экран				

## 6.Текст программы на языке Ассемблера

```

1          ; Работу выполнил Никулин ИУ5-41Б Лаборатрная работа №3
2
3 0000      MYCODE SEGMENT 'CODE'
4          ASSUME      CS:MYCODE, DS:MYCODE
5 0000 4E      LET DB 'N'
6 0001      START:
7          ; Загрузка      сегментного регистра данных DS
8 0001 0E      PUSH CS
9 0002 1F      POP DS
10         ; Вывод одного символа на экран
11 0003 E8 0024      CALL PRINTLET
12 0006 E8 002A      CALL CRLF
13 0009 80 06 0000r 01      ADD LET, 1
14         ; Вывод второго символа на      экран
15 000E E8 0019      CALL PRINTLET
16 0011 E8 001F      CALL CRLF
17 0014 80 06 0000r 01      ADD LET, 1
18         ; Вывод третьего символа на экран
19 0019 E8 000E      CALL PRINTLET
20 001C E8 0014      CALL CRLF
21 001F 80 06 0000r 01      ADD LET, 1
22         ; Выход из      программы
23 0024 B0 00      MOV AL, 0
24 0026 B4 4C      MOV AH, 4CH
25 0028 CD 21      INT 21H
26
27 002A      PRINTLET PROC NEAR
28 002A B4 02      MOV AH, 02
29 002C 8A 16 0000r      MOV DL, LET
30 0030 CD 21      INT 21H
31 0032 C3      RETN
32 0033      PRINTLET ENDP
33

```

34	0033		CRLF PROC NEAR
35	0033 B4 02		MOV AH, 02
36	0035 B2 0A		MOV DL, 10
37	0037 CD 21		INT 21H
38	0039 B4 02		MOV AH, 02
39	003B B2 0D		MOV DL, 13
40	003D CD 21		INT 21H
41	003F C3		RETN
42	0040		CRLF ENDP
43			
44	0040		MYCODE ENDS
45			END START

Symbol Name	Type	Value	Cref (defined at #)
??DATE	Text	"03/27/23"	
??FILENAME	Text	"FIRSTD "	
??TIME	Text	"11:06:11"	
??VERSION	Number	030A	
@CPU	Text	0101H	
@CURSEG	Text	MYCODE	#3
@FILENAME	Text	FIRSTD	
@WORDSIZE	Text	2	#3
CRLF	Near	MYCODE:0033	12 16 20 #34
LET	Byte	MYCODE:0000	#5 13 17 21 29
PRINTLET	Near	MYCODE:002A	11 15 19 #27
START	Near	MYCODE:0001	#6 45

Groups & Segments	Bit	Size	Align	Combine	Class	Cref (defined at #)
MYCODE	16	0040	Para	none	CODE	#3 4 4

## 7. Результаты работы программы

U:\TASM3>FIRSTD N O P  U:\TASM3>	U:\TASM3			
	Name	Name		Name
..	..	firstd	lst	tasm hlp
asmed ini	firstd	map	tasm_exm	arj
asm_ed exe	firstd	obj	td	exe
asm_edit zip	grep	com	td	hlp
choice32 exe	hello	asm	tdconfig	td
default shk	hello	exe	tlink	exe
dpmi16bi ovl	hello	map	tlink	hlp
dpmi32um ovl	hello	obj	trans	exe
dpmi1oad exe	lab3	asm	vc	com
dpmimem dll	lab4	asm	vc	ini
first asm	lab4	exe		
first exe	lab4	lst		
first lst	lab4	map		
first map	lab4	obj		
first obj	mem	exe		
first txt	rkm	com		
firstd asm	run	bat		
firstd exe	tasm	exe		
firstd.lst		2425	3-27-23	11:06a

## 8. Выводы по ЛР № 3

Мы разработали и провели отладку программы на языке Ассемблер, изучили способ написания текста программы, выполнения ее компиляции, редактирования связей и компоновки исполняемого модуля программы.