

Защищено:
Большаков С.А.

Демонстрация ЛР:
Большаков С.А.

"__" _____ 2024 г.

"__" _____ 2024 г.

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

"Вывод трех символов"

(есть ли дополнительные требования - ДА)

6
(количество листов)
Вариант № 11

ИСПОЛНИТЕЛЬ:

студент группы **ИУ5-41Б**

Ларкин Б. В.

(подпись)

"__" _____ 2024 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. Текст программы на языке Ассемблера (.LST).....	4
6. Скриншот программы в TD.exe.....	6
7. Результаты работы программы.....	6
8. Выводы по ЛР № 3.....	6

1. Цель выполнения лабораторной работы № 3

Лабораторная работа №3 выполняется для получения навыков разработки и отладки программ на ЯП, получения базовых знаний об использовании прерываний, процедур и регистров на Ассемблере, изучения и использования компонентов системы программирования Ассемблер (компилятора, редактора связей, отладчика) и получения навыков оформления документации по программным разработкам, реализуемым на языке.

2. Порядок и условия проведения работы № 3

Разработать и отладить программу на языке Ассемблер для вывода на экран дисплея трех первых заглавных русских букв (А, Б, В), на трех отдельных строках дисплея подряд (отдельно программируется перевод строки и возврат каретки!).

После завершения вывода букв на экран организовать ожидание ввода любого символа с клавиатуры (нажатие клавиши).

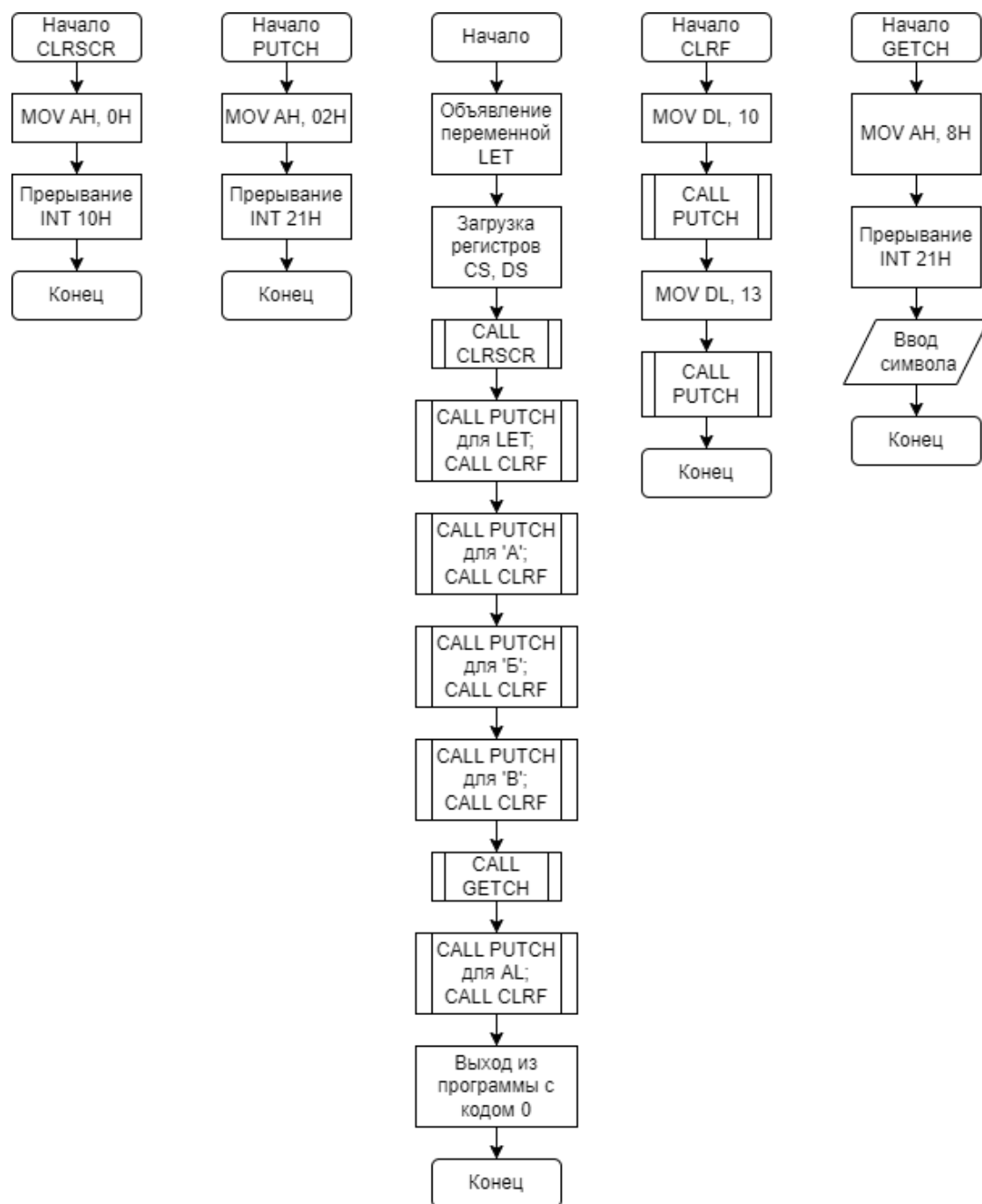
Необходимо использовать процедуры при разработке программы. Предусмотреть минимально три процедуры: для ввода символа (1-я процедура - **GETCH** название процедуры ввода символа желательно взять такое название), для вывода одного символа (2-я процедура - **PUTCH**) и для перевода строки с возвратом каретки (3-я процедура - **CLRF**) на дисплее (оформление процедур - **PROC - ENDP**, вызов процедур - **CALL**).

В программе организовать очистку экрана до начала вывода символов, а также после завершения работы программы. Очистка экрана должна выполняться отдельной дополнительной процедурой на языке Ассемблер (название ее - **CLRSCR**). Очистка экрана должна быть выполнена без организации циклов вывода символов с помощью соответствующего прерывания (найденного вами в справочнике). При выполнении дополнительных требований в текст программы добавляется специальный комментарий, подтверждающий их выполнение. На титульном листе отчета нужно отметить факт выполнения ЛР с дополнительными требованиями.

3. Описание ошибок, возникших при отладке № 3

№ п/п	Проявление ошибки	Причина ошибки	Способ устранения
1.	Некорректное получение символа, набранного с клавиатуры	Указание в теле процедуры GETCH вместо регистра АН регистра AL	Замена регистра: MOV AH, 08H

4. Блок-схема программы



5. Текст программы на языке Ассемблера (.LST)

Turbo Assembler	Version 3.1	03/24/24 21:14:59	firstd.asm
1	0000	MYCODE SEGMENT 'CODE'	
2		;JP#3 2024 ЛЯРКИН ИУ5-41Б Вар#11	
3		ASSUME CS:MYCODE	
4		PUBLIC LET	
5	0000 8B	LET DB 'Л'	
6			
7		;Дополнительное задание	
8	0001	CLRSCR PROC	
9	0001 B4 00	MOV AH, 0H	
10	0003 CD 10	INT 10H	
11	0005 C3	RET	
12	0006	CLRSCR ENDP	

13		
14	0006	PUTCH PROC
15	0006 B4 02	MOV AH, 02H
16	0008 CD 21	INT 21H
17	000A C3	RET
18	000B	PUTCH ENDP
19		
20		;Кладет полученный символ в AL
21	000B	GETCH PROC
22	000B B4 08	MOV AH, 08H
23	000D CD 21	INT 21H
24	000F C3	RET
25	0010	GETCH ENDP
26		
27	0010	CLRF PROC
28	0010 B2 0A	MOV DL, 10
29	0012 E8 FFF1	CALL PUTCH
30	0015 B2 0D	MOV DL, 13
31	0017 E8 FFEC	CALL PUTCH
32	001A C3	RET
33	001B	CLRF ENDP
34		
35	001B	START:
36		; Загрузка сегментного регистра данных DS
37	001B 0E	PUSH CS
38	001C 1F	POP DS
39		; Вывод символов на экран
40	001D E8 FFE1	CALL CLRSCR
41	0020 2E: 8A 16	0000r MOV DL, LET
42	0025 E8 FFDE	CALL PUTCH
43	0028 E8 FFE5	CALL CLRF
44	002B B2 80	MOV DL, 'A'
45	002D E8 FFD6	CALL PUTCH
46	0030 E8 FFDD	CALL CLRF
47	0033 B2 81	MOV DL, 'Б'
48	0035 E8 FFCE	CALL PUTCH
49	0038 E8 FFD5	CALL CLRF
50	003B B2 82	MOV DL, 'В'
51	003D E8 FFC6	CALL PUTCH
52	0040 E8 FFCD	CALL CLRF
53	0043 E8 FFC5	CALL GETCH
54	0046 8A D0	MOV DL, AL
55	0048 E8 FFBB	CALL PUTCH
56	004B E8 FFC2	CALL CLRF
57		; Выход из программы
58	004E B0 00	MOV AL, 0
59	0050 B4 4C	MOV AH, 4CH
60	0052 CD 21	INT 21H
61	0054	MYCODE ENDS
62		END START

6. Скриншот программы в TD.exe

The screenshot shows the TD debugger interface. The left pane displays assembly code for a module named 'firstd'. The code includes a segment definition, a public label 'LET', and two procedures: 'CLRSCR' and 'PUTCH'. The right pane shows the CPU 80486 registers and memory dump. The registers are initialized to zero, except for 'ip' which is 001B. The memory dump shows the current instruction address and the next instruction address.

CPU 80486	
#firstd#start	ax 0000 c=0
cs:001B PUSH CS	bx 0000 z=0
cs:001C POP DS	cx 0000 s=0
cs:001D CALL CLRSC	dx 0000 o=0
cs:0020 MOV DL, LE	si 0000 p=0
cs:0025 CALL PUTCH	di 0000 a=0
cs:0028 CALL CLRSC	bp 0000 i=1
cs:002B MOV DL, 'A	sp 0000 d=0
cs:002D CALL PUTCH	ds 4EE9
cs:0030 CALL CLRSC	es 4EE9
cs:0033 MOV DL, 'B	ss 4EF8
cs:0035 CALL PUTCH	cs 4EF9
cs:0038 CALL CLRSC	ip 001B
cs:003B MOV DL, 'B	
cs:003D CALL PUTCH	
ds:0000 CD 20 FF 9F	ss:0002 6474
ds:0008 AD DE E0 01	ss:0000 0000
ds:0010 11 1C 89 02	ss:FFFE FFFF
ds:0018 01 01 01 00	ss:FFFC 0000
ds:0020 FF FF FF FF	ss:FFFA 0000

TD.exe Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu

7. Результаты работы программы

С клавиатуры был введен символ 'S'.



8. Выводы по ЛР № 3

Разработан файл .ASM и соответствующие файлы приложения и листинга на языке Ассемблер. Программа корректно выводит единичные символы при помощи перемещений данных в регистрах, использования прерываний системы. Так мы изучили основы языка Ассемблер.