

Защищено:
Большаков С.А.

Демонстрация ЛР:
Большаков С.А.

"__" _____ 2024 г.

"__" _____ 2024 г.

**Отчет по лабораторной работе № 7 по курсу
Системное программирование**

"Ввод, вывод и перевод адреса"

(есть ли дополнительные требования - ДА)

10
(количество листов)
Вариант № 20

ИСПОЛНИТЕЛЬ:

студент группы ИУ5-41Б

Цыпышев Т.А.

(подпись)

"__" _____ 2024 г.

СОДЕРЖАНИЕ

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

Цель выполнения лабораторной работы № 7

Разработать и отладить программу на языке Ассемблер для ввода с клавиатуры четырехразрядного шестнадцатеричного числа – символами! (короткого адреса NEAR) в машинное шестнадцатеричное представление (доступные шестнадцатеричные цифры – 0123456789ABCDEF). Введенное значение переводиться в машинное представление в виде отдельного слова (2 байта – DW – тип переменной). Полученное значение выводится затем на экран также в шестнадцатеричном представлении, но заново переведенное из машинного формата. Кроме того, выполняется перевод по схеме Горнера (см. в Википедии) в десятичное представление и на экран выводится в десятичном формате (нужно выполнить программный перевод из одной системы счисления в другую).

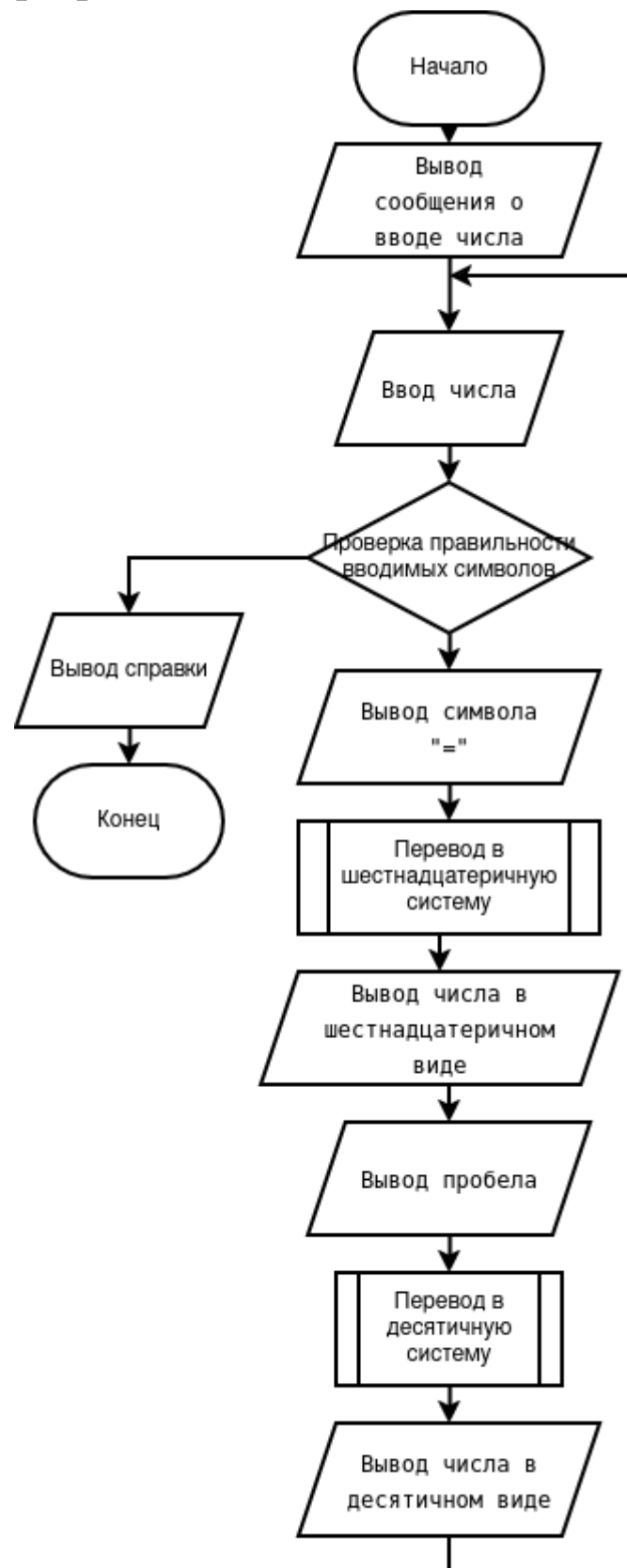
Порядок и условия проведения работы № 7

1. **Ввод чисел и вывод их представлений:** Программа должна принимать ввод чисел в шестнадцатеричном формате и выводить их шестнадцатеричное и десятичное представления, разделенные знаком равенства ("="). Между шестнадцатеричным и десятичным представлениями должен быть пробел.
2. **Организация цикла ввода:** Программа должна работать в циклическом режиме, запросив ввод нового числа после завершения ввода предыдущего. Завершение цикла ввода чисел происходит при вводе символа "*" в первой позиции строки ввода.
3. **Проверка вводимых символов:** Необходимо проверять вводимые шестнадцатеричные символы на корректность (допустимые символы: 0-9 и A-F).
4. **Очистка экрана и сообщение об успешном завершении:** Программа должна начинаться с очистки экрана и завершаться сообщением об успешном завершении, а также выводом данных студента: ФИО, группы и номера варианта.
5. **Подсказка для ввода чисел:** Перед запросом вводимого числа должна выдаваться подсказка в виде: "Введите число(длинный адрес: NNNN:NNNN)>"

Описание ошибок, возникших при отладке № 7

№ п/п	Проявление ошибки	Причина ошибки	Способ устранения
1.	Неправильный перевод в десятичную систему счисления	Не возможность нормально использовать уже готовую функцию HEX	Была написана новая функция HEX_TO_DECIMAL

Блок-схема программы



Текст программы на языке Ассемблера (.LST)

Turbo Assembler Version 3.1 05/08/24 05:10:39 Page 1
lab.asm

```
1 ; Лабораторная работа №7
2 ; Цыпышев Т.А. ИУ5-41 Вар. 20
3
=====
4
5 0000 prgr SEGMENT
6 ASSUME CS:prgr
7
8 0000 BEGIN:
9 0000 E9 0167 JMP MAIN
10
11 ; Сообщение о помощи
12 0003 96 EB AF EB E8 A5 A2+ help_message DB 'Цыпышев Тимофей ИУ5-41Б Варю 20$'
13 20 92 A8 AC AE E4 A5+
14 A9 20 88 93 35 2D 34+
15 31 81 20 82 A0 E0 EE+
16 20 32 30 24
17 0023 0100*(00) buffer DB 256 DUP(0)
18 0123 00 flag_var DB 0
19 0124 82 A2 A5 A4 A8 E2 A5+ input_message DB 'Введите число (длинный адрес:
HHHH:HHHH)> '$
20 20 E7 A8 E1 AB AE 20+
21 28 A4 AB A8 AD AD EB+
22 A9 20 A0 A4 E0 A5 E1+
23 3A 20 8D 8D 8D 8D 3A+
24 8D 8D 8D 8D 29 3E 20+
25 24
26 014F 2710 03E8 0064 000A + digits DW 10000,1000,100,10,1
27 0001
28 0159 30 31 32 33 34 35 36+ hex_digits DB '0123456789ABCDEF$'
29 37 38 39 41 42 43 44+
30 45 46 24
31
32 016A MAIN PROC
33 016A E8 010F CALL CLRSCR
34 016D main_loop:
35 016D B4 09 MOV AH,09H ; Вывод сообщения о
вводе числа
36 016F BA 0124r MOV DX,OFFSET input_message
37 0172 CD 21 INT 21H
38
39 0174 E8 0034 call INPUT ; Ввод числа
40 0177 2E: 80 3E 0123r 01 cmp flag_var,1
41 017D 74 1C je main_exit
42 017F 53 push BX
43
44 0180 B2 3D mov dl,'=' ; Вывод символа "="
45 0182 E8 00DD call PUTC
46
47 0185 8B D3 mov dx,bx ; Вывод числа в
шестнадцатеричном виде
48 0187 E8 00A8 call HEX_WRITE
49
50 018A B2 20 mov dl,' ' ; Вывод пробела
51 018C E8 00D3 call PUTC
52
53 018F 5B pop BX
54 0190 E8 00AE call HEX_TO_DECIMAL ; Вывод числа в
десятичном виде
```

55				
56	0193 E8 00DB		call CLRF	; Переход на новую строку
57	0196 B9 0002		MOV CX,2	; Оставшееся количество
итераций				
Turbo Assembler Version 3.1 05/08/24 05:10:39 Page 2				
lab.asm				
58	0199 E2 D2		LOOP	main_loop
59				
60	019B	main_exit:		
61	019B B4 09		MOV AH,09H	; Вывод сообщения о
помощи				
62	019D BA 0003r		MOV DX,OFFSET help_message	
63	01A0 CD 21		INT 21H	
64	01A2 E8 00C2		CALL GETCH	
65	01A5 E8 00D4		CALL CLRSCR	
66	01A8 E8 00D8		CALL QUIT_PROGRAM	
67	01AB	MAIN ENDP		
68				
69		; Процедура ввода числа		
70	01AB	INPUT PROC		
71	01AB BD 0000		mov bp,0	; Инициализация счетчика
72	01AE BB 0000		mov bx,0	; Инициализация результата
73	01B1	in_loop:		
74	01B1 E8 00B8		call GETCH_NON_BLOCKING	; Получение
символа				
75	01B4 8A D0		mov dl,al	
76	01B6 EB 19 90		jmp check	
77				
78	01B9	shifting:		
79	01B9 E8 00A6		call PUTCH	; Вывод символа
80	01BC 58	pop	ax	; Получение символа из стека
81	01BD B4 00		mov ah,0	
82	01BF D1 E3 D1 E3 D1 E3 D1+		shl bx, 4	; Сдвиг результата на 4 бита
влево				
83	E3			
84				
85	01C7 03 D8		add bx,ax	; Добавление новой цифры к
результату				
86	01C9 45	inc	bp	; Инкремент счетчика
87	01CA 83 FD 04		cmp bp,4	; Проверка количества
введенных цифр				
88	01CD 74 36		je exit	; Если введены 4 цифры, выход
89	01CF EB E0		jmp in_loop	
90				
91	01D1	check:		
92	01D1 3C 2A		cmp al,'*'	; Проверка на символ
завершения ввода				
93	01D3 75 09		jne number	; Если не "*", обработка чисел
94	01D5 2E: C6 06 0123r 01	mov	flag_var,1	; Установка флага завершения ввода
95	01DB EB 28 90		jmp exit	
96				
97	01DE	number:		
98	01DE 3C 30		cmp al,'0'	; Проверка на цифру
99	01E0 72 09		jb big_case	; Если меньше "0", переход к
обработке больших +				
100		символов		
101				
102	01E2 3C 39		cmp al,'9'	; Проверка на цифру
103	01E4 77 05		ja big_case	; Если больше "9", переход к
обработке больших +				
104		символов		
105				
106	01E6 2C 30		sub al,'0'	; Преобразование символа в число

107	01E8 50		push ax	; Запись числа в стек
108	01E9 EB CE		jmp	shifting ; Переход к сдвигу
109				
110	01EB	big_case:		
"F"	111 01EB 3C 41		cmp	al,'A' ; Проверка на символ от "A" до
	112 01ED 72 09		jb	small_case ; Если меньше "A",
переход к обработке маленьких +	113	СИМВОЛОВ		
	114			
Turbo Assembler Version 3.1	05/08/24 05:10:39		Page 3	
lab.asm				
115	01EF 3C 46		cmp	al,'F' ; Проверка на символ от "A" до
"F"	116 01F1 77 05		ja	small_case ; Если больше "F",
переход к обработке маленьких +	117	СИМВОЛОВ		
	118			
число	119 01F3 2C 37		sub	al,'A'-10 ; Преобразование символа в
	120 01F5 50		push ax	; Запись числа в стек
	121 01F6 EB C1		jmp	shifting ; Переход к сдвигу
	122			
	123 01F8	small_case:		
"f"	124 01F8 3C 61		cmp	al,'a' ; Проверка на символ от "a" до
	125 01FA 72 B5		jb	in_loop ; Если меньше "a",
продолжение ввода	126			
"f"	127 01FC 3C 66		cmp	al,'f' ; Проверка на символ от "a" до
	128 01FE 77 B1		ja	in_loop ; Если больше "f",
продолжение ввода	129			
число	130 0200 2C 57		sub	al,'a'-10 ; Преобразование символа в
	131 0202 50		push ax	; Запись числа в стек
	132 0203 EB B4		jmp	shifting ; Переход к сдвигу
	133			
	134 0205	exit:		
	135 0205 C3		ret	
	136 0206	INPUT ENDP		
	137			
	138	; Процедура вывода числа в шестнадцатеричном виде		
	139 0206	HEX PROC		
	140 0206 1E		push ds	
	141 0207 50		push ax	
	142 0208 53		push bx	
	143 0209 51		push cx	
	144 020A 0E		push cs	
	145 020B 1F		pop ds	
	146			
	147 020C BB 0159r		lea	bx,hex_digits
	148			
	149 020F 8A C6		mov	al,dh
	150 0211 D0 E8 D0 E8 D0 E8 D0+		shr	al, 4
	151 E8			
	152 0219 D7		xlat	
	153 021A 52		push dx	
	154 021B 8A D0		mov	dl,al
	155 021D B4 02		mov	ah,02h
	156 021F CD 21		int	21h
	157			

```

158 0221 8A C6          mov     al,dh
159 0223 24 0F          and     al,00001111b
160 0225 D7             xlat
161 0226 8A D0          mov     dl,al
162 0228 B4 02          mov     ah,02h
163 022A CD 21          int     21h
164
165 022C 5A             pop     dx
166 022D 59             pop     cx
167 022E 5B             pop     bx
168 022F 58             pop     ax
169 0230 1F             pop     ds
170 0231 C3             ret
171 0232                HEX ENDP
Turbo Assembler   Version 3.1   05/08/24 05:10:39   Page 4
lab.asm

```

```

172
173                ; Процедура вывода числа в шестнадцатеричном виде, с
последующим "h"
174 0232                HEX_WRITE PROC
175 0232 52             push    dx
176 0233 E8 FFD0        call    HEX
177 0236 5A             pop     dx
178 0237 8A F2          mov     dh,dl
179 0239 E8 FFCA        call    HEX
180 023C B2 68          mov     dl,'h'
181 023E CD 21          int     21h
182 0240 C3             ret
183 0241                HEX_WRITE ENDP
184
185                ; Процедура преобразования шестнадцатеричного числа в десятичное
186 0241                HEX_TO_DECIMAL PROC
187 0241 8B C3          mov     ax,bx
188 0243 BF 0000        mov     di,0
189 0246                hex_loop:
190 0246 BA 0000        mov     dx,0
191 0249 2E: 8B 9D    014Fr  mov     bx,digits[di]
192 024E F7 F3          div     bx
193 0250 52             push    dx
194 0251 05 0030        add     ax,'0'
195 0254 8A D0          mov     dl,al
196 0256 E8 0009        call    PUTCH
197 0259 58             pop     ax
198 025A 47             inc     di
199 025B 47             inc     di
200 025C 83 FF 0A        cmp     di,10
201 025F 72 E5          jb     hex_loop
202 0261 C3             RET
203 0262                HEX_TO_DECIMAL ENDP
204
205                ; Процедура вывода символа
206 0262                PUTCH PROC
207 0262 B4 02          mov     ah,2
208 0264 CD 21          int     21h
209 0266 C3             ret
210 0267                PUTCH ENDP
211
212                ; Процедура получения символа
213 0267                GETCH PROC
214 0267 B4 01          mov     ah,1
215 0269 CD 21          int     21h
216 026B C3             ret
217 026C                GETCH ENDP

```



```

218
219 ; Процедура получения символа без блокировки
220 026C GETCH_NON_BLOCKING PROC
221 026C B4 08 mov ah,8
222 026E CD 21 int 21h
223 0270 C3 ret
224 0271 GETCH_NON_BLOCKING ENDP
225
226 ; Процедура перехода на новую строку
227 0271 CLRF PROC
228 0271 B2 0A mov dl,10
Turbo Assembler Version 3.1 05/08/24 05:10:39 Page 5
lab.asm

```

```

229 0273 E8 FFEC call PUTCH
230 0276 B2 0D mov dl,13
231 0278 E8 FFE7 call PUTCH
232 027B C3 ret
233 027C CLRF ENDP
234
235 ; Процедура очистки экрана
236 027C CLRSCR PROC
237 027C B4 00 mov ah,00
238 027E B0 03 mov al,03
239 0280 CD 10 int 10h
240 0282 C3 RET
241 0283 CLRSCR ENDP
242
243 ; Процедура завершения программы
244 0283 QUIT_PROGRAM PROC
245 0283 B4 4C mov ah,4ch
246 0285 B0 00 mov al,00h
247 0287 CD 21 int 21h
248 0289 QUIT_PROGRAM ENDP
249
250 0289 prgr ENDS
251 END BEGIN

```

Turbo Assembler Version 3.1 05/08/24 05:10:39 Page 6
Symbol Table

Symbol Name	Type	Value	Cref (defined at #)
??DATE	Text	"05/08/24"	
??FILENAME	Text	"lab "	
??TIME	Text	"05:10:38"	
??VERSION	Number	030A	
@CPU	Text	0101H	
@CURSEG	Text	PRGR	#5
@FILENAME	Text	LAB	
@WORDSIZE	Text	2	#5
BEGIN	Near	PRGR:0000	#8 251
BIG_CASE	Near	PRGR:01EB	99 103 #110
BUFFER	Byte	PRGR:0023	#17
CHECK	Near	PRGR:01D1	76 #91
CLRF	Near	PRGR:0271	56 #227
CLRSCR	Near	PRGR:027C	33 65 #236
DIGITS	Word	PRGR:014F	#26 191
EXIT	Near	PRGR:0205	88 95 #134
FLAG_VAR	Byte	PRGR:0123	#18 40 94
GETCH	Near	PRGR:0267	64 #213
GETCH_NON_BLOCKING	Near	PRGR:026C	74 #220

HELP_MESSAGE	Byte	PRGR:0003	#12	62
HEX	Near	PRGR:0206	#139	176 179
HEX_DIGITS	Byte	PRGR:0159	#28	147
HEX_LOOP	Near	PRGR:0246	#189	201
HEX_TO_DECIMAL	Near	PRGR:0241	54	#186
HEX_WRITE	Near	PRGR:0232	48	#174
INPUT	Near	PRGR:01AB	39	#70
INPUT_MESSAGE	Byte	PRGR:0124	#19	36
IN_LOOP	Near	PRGR:01B1	#73	89 125 128
MAIN	Near	PRGR:016A	9	#32
MAIN_EXIT	Near	PRGR:019B	41	#60
MAIN_LOOP	Near	PRGR:016D	#34	58
NUMDER	Near	PRGR:01DE	93	#97
PUTCH	Near	PRGR:0262	45 51 79 196	#206 229 231
QUIT_PROGRAM	Near	PRGR:0283	66	#244
SHIFTING	Near	PRGR:01B9	#78	108 121 132
SMALL_CASE	Near	PRGR:01F8	112	116 #123

Groups & Segments Bit Size Align Combine Class Cref (defined at #)

PRGR 16 0289 Para none #5 6

Результаты работы программы

```
Введите число (длинный адрес: HHHH:HHHH)> 00fe=00FEh 00254
```

```
Введите число (длинный адрес: HHHH:HHHH)> _
```

Выводы по ЛР № 7

Изучив задание по разработке программы на языке Ассемблер для ввода, преобразования и вывода шестнадцатеричного числа, я усвоил следующее:

1. Освоил методы ввода данных с клавиатуры в программе на языке Ассемблер.
2. Понял процесс преобразования символов, введенных с клавиатуры, в машинное шестнадцатеричное представление.
3. Изучил принципы работы со строками символов и их преобразования в числовые значения.
4. Получил опыт работы с типом переменной DW (2 байта) для хранения преобразованных числовых значений.
5. Освоил алгоритм перевода числа по схеме Горнера для получения десятичного представления шестнадцатеричного числа.
6. Получил навыки работы с выводом данных на экран в шестнадцатеричном и десятичном форматах.
7. Осознал важность отладки программы для обеспечения правильного преобразования и вывода чисел.