Московский государственный технический университет им. Н.Э.Баумана

|  |  |  |
| --- | --- | --- |
| Защищено:  Большаков С.А.  "\_\_"\_\_\_\_\_\_\_\_\_\_\_\_\_2024 г. |  | Демонстрация ЛР:  Большаков С.А.  "\_\_"\_\_\_\_\_\_\_\_\_\_\_\_\_2024 г. |

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

**Системное программирование**

#### "Ввод, вывод и перевод адреса"

#### (есть ли дополнительные требования - НЕТ)

8

(количество листов)

Вариант № 11

|  |  |
| --- | --- |
| ИСПОЛНИТЕЛЬ: |  |
| студент группы **ИУ5-41Б** | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
|  | (подпись) |
| **Ларкин Б. В.** | "\_\_"\_\_\_\_\_\_\_\_\_\_\_\_\_2024 г. |

Москва, МГТУ - 2024

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**СОДЕРЖАНИЕ**

[1. Цель выполнения лабораторной работы № 5 3](#_Toc165383330)

[2. Порядок и условия проведения работы № 5 3](#_Toc165383331)

[3. Описание ошибок, возникших при отладке № 5 3](#_Toc165383332)

[4. Блок-схема программы 4](#_Toc165383333)

[5. Текст программы на языке Ассемблера (.LST) 4](#_Toc165383334)

[6. Скриншот программы в TD.exe 8](#_Toc165383335)

[7. Результаты работы программы 8](#_Toc165383336)

[8. Выводы по ЛР № 5 8](#_Toc165383337)

# Цель выполнения лабораторной работы № 7

Лабораторная работа №7 выполняется для получения навыков разработки программ и процедур на Ассемблере, использующих адреса и методы их ввода, вывода и перевода.

# Порядок и условия проведения работы № 7

Разработать и отладить программу на языке Ассемблер для ввода с клавиатуры четырехразрядного шестнадцатеричного числа – символами! (короткого адреса NEAR) в машинном шестнадцатеричном представлении (доступные шестнадцатеричные цифры – 0123456789АBCDEF). 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).

Между введенным символьным значением адреса и выводимым шестнадцатеричным представлением должен располагаться знак равенства ("="), а между – формируемыми представлениями пробел (шестнадцатеричным и десятичным).

Например (сначала машинное - 00FEh ,а затем десятичное - 254):

Введите число( длинный адрес: НННН:НННН)>00FE=00FEh 254

>...

>\*

Завершение ввода чисел!

Программа должна работать в циклическом режиме, то есть после ввода одного числа, запрашивается ввод нового. Завершение цикла ввода чисел выполняется по знаку “\*” в первой позиции строки ввода. Для ввода и перевода должны быть использованы базовые процедуры (см. ЛР выше). При вводе необходимо проверять вводимые шестнадцатеричные символы (0-9 и A -F). Нужно организовать очистку экрана до начала работы программы, и после ее завершения. По завершению программы выдается сообщение об ее успешном окончании и данные студента: ФИО, группа и номер варианта. Для запроса вводимого числа предварительно должна выдаваться подсказка в виде:

”Введите число( длинный адрес: НННН:НННН)>”:

Оформить отчет по ЛР. Для оформления отчета студент должен знать или найти способ для вывода результата работы программы в текстовый файл. Лучше использовать копирование текста из окна командной строки (нежелательно снимать графическую картинку с экрана). Программа может быть выполнена в виде \*.EXE исполнимого модуля.

# Описание ошибок, возникших при отладке ЛР № 7

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Проявление ошибки | Причина ошибки | Способ устранения |
|  | Вход в бесконечный цикл при вводе символов не 16-й СС | Зацикливание меток внутри цикла ввода и проверки символов на валидность | Сравнение введенного символа с каждым из числа возможных, прыжок на др. метку |
|  | Раннее завершение программы при вводе некорректного символа | Прыжок на метку MEND после ввода неверного символа | Добавление метки в начало функции, прыжок на нее из метки ошибочного ввода. |

# Блок-схема программы

# 5. Текст программы на языке Ассемблера (.LST)

Turbo Assembler Version 3.1 05/02/24 22:22:46

laba7.asm

1 ;ЛР #7 2024 Ларкин ИУ5-41Б

2 0000 DATASG SEGMENT 'DATA'

3 0000 30 31 32 33 34 35 36+ TABLHEX DB '0123456789ABCDEF'

4 37 38 39 41 42 43 44+

5 45 46

6 0010 82 A2 A5 A4 A8 E2 A5+ MSG DB 'Введите шестнадцатеричное число(HHHH, \* - конец программы):$'

7 20 E8 A5 E2 AD A0 A4+

8 E6 A0 E2 A5 E0 A8 E7+

9 AD AE A5 20 E7 A8 E1+

10 AB AE 28 48 48 48 48+

11 2C 20 2A 20 2D 20 AA+

12 AE AD A5 E6 20 AF E0+

13 AE A3 E0 A0 AC AC EB+

14 29 3A 24

15 004B 64\*(00) BUF DB 100 DUP( 0 )

16 00AF 0000 DECW DW 0

17 00B1 8E E8 A8 A1 AA A0 20+ MSGERR DB 'Ошибка символа!$'

18 E1 A8 AC A2 AE AB A0+

19 21 24

20 00C1 DATASG ENDS

21

22 0000 STSEG SEGMENT STACK 'STACK'

23 0000 0100\*(0000) DW 256 DUP(0)

24 0200 STSEG ENDS

25

26 0000 MYCODE SEGMENT 'CODE'

27 ASSUME CS:MYCODE, DS:DATASG, SS:STSEG

28 0000 START:

29 ; Загрузка сегментного регистра данных DS

30 0000 B8 0000s MOV AX, DATASG

31 0003 8E D8 MOV DS, AX

32 ;; Цикл проверки ввода \*

33 0005 B4 00 MOV AH, 00H

34 0007 B0 03 MOV AL, 03H

35 0009 CD 10 INT 10H

36 ;

37 000B B4 09 MOV AH, 9H

38 000D BA 0010r LEA DX, MSG

39 0010 CD 21 INT 21h

40 0012 E8 013A CALL LFCR

41 ;;

42 0015 B9 000A MOV CX, 10

43 0018 METLOOP:

44 ; Ввод шетн. числа и запоминание в BUF

45 0018 E8 0030 CALL HEXADR

46 ;

47 001B B2 20 MOV DL, ' '

48 001D E8 012A CALL PUTCH

49 0020 B2 3D MOV DL, '='

50 0022 E8 0125 CALL PUTCH

51 0025 B2 20 MOV DL, ' '

52 0027 E8 0120 CALL PUTCH

53 ; Вывод шестнадцатеричного числа

54 002A E8 00A7 CALL PRINTHEX

55 002D B2 20 MOV DL, ' '

56 002F E8 0118 CALL PUTCH

57 0032 B2 20 MOV DL, ' '

58 0034 E8 0113 CALL PUTCH

59 ; Перевод в десятичное и печать

60 0037 E8 00BB CALL DECPRINT

61 ; Цикл вывода шетнадцатеричной строки

62 003A E8 0112 CALL LFCR

63 003D E2 D9 LOOP METLOOP

64

65 003F MEND:

66 ; Очистка экрана

67 003F B4 00 MOV AH, 00H

68 0041 B0 03 MOV AL, 03H

69 0043 CD 10 INT 10H

70

71 ; Выход из прораммы

72 0045 B4 4C MOV AH, 4Ch

73 0047 B0 00 MOV AL, 0

74 0049 CD 21 INT 21H

75 ; Продецуры программы

76 004B HEXADR PROC

77 ; Подготовка цикла ввода

78 004B proc\_start:

79 004B BE 004Br MOV SI , OFFSET BUF

80 004E B9 0004 MOV CX , 4

81 ; Цикл до 4-х символов

82 0051 MVVOD:

83 0051 MCICL:

84 0051 E8 0106 CALL GETSIMB

85 0054 3C 2A CMP AL, '\*'

86 0056 74 E7 JE MEND

87 0058 MC1:

88 ; Проверка символа на правильность

89 0058 3C 30 CMP AL , 30H

90 005A 74 6A JE MBUF

91 005C 3C 31 CMP AL , 31H

92 005E 74 66 JE MBUF

93 0060 3C 32 CMP AL , 32H

94 0062 74 62 JE MBUF

95 0064 3C 33 CMP AL , 33H

96 0066 74 5E JE MBUF

97 0068 3C 34 CMP AL , 34H

98 006A 74 5A JE MBUF

99 006C 3C 35 CMP AL , 35H

100 006E 74 56 JE MBUF

101 0070 3C 36 CMP AL , 36H

102 0072 74 52 JE MBUF

103 0074 3C 37 CMP AL , 37H

104 0076 74 4E JE MBUF

105 0078 3C 38 CMP AL , 38H

106 007A 74 4A JE MBUF

107 007C 3C 39 CMP AL , 39H

108 007E 74 46 JE MBUF

109 0080 3C 41 CMP AL , 'A'

110 0082 74 42 JE MBUF

111 0084 3C 42 CMP AL , 'B'

112 0086 74 3E JE MBUF

113 0088 3C 43 CMP AL, 'C'

114 008A 74 3A JE MBUF

115 008C 3C 44 CMP AL , 'D'

116 008E 74 36 JE MBUF

117 0090 3C 45 CMP AL , 'E'

118 0092 74 32 JE MBUF

119 0094 3C 46 CMP AL, 'F'

120 0096 74 2E JE MBUF

121 0098 3C 61 CMP AL , 'a'

122 009A 74 2A JE MBUF

123 009C 3C 62 CMP AL , 'b'

124 009E 74 26 JE MBUF

125 00A0 3C 63 CMP AL, 'c'

126 00A2 74 22 JE MBUF

127 00A4 3C 64 CMP AL , 'd'

128 00A6 74 1E JE MBUF

129 00A8 3C 65 CMP AL , 'e'

130 00AA 74 1A JE MBUF

131 00AC 3C 66 CMP AL, 'f'

132 00AE 74 16 JE MBUF

133

134 00B0 ERROR:

135 00B0 8A D0 MOV DL, AL

136 00B2 E8 0095 CALL PUTCH

137 00B5 E8 0097 CALL LFCR

138 00B8 B0 23 MOV AL,'#'

139 00BA BA 00B1r MOV DX , OFFSET MSGERR

140 00BD B4 09 mov AH , 09H

141 00BF CD 21 INT 21H

142 00C1 E8 008B CALL LFCR

143 00C4 EB 85 JMP proc\_start

144

145 ; Запись в буфер и печать

146 00C6 MBUF:

147 00C6 88 04 MOV [SI], AL

148 00C8 46 INC SI

149 ; Печать символа

150 00C9 8A D0 MOV DL, AL

151 00CB E8 007C CALL PUTCH

152 00CE E2 81 LOOP MVVOD

153 ;

154 00D0 C6 04 24 MOV BYTE PTR [SI], '$'

155 00D3 C3 RET

156 00D4 HEXADR ENDP

157 ;;

158 00D4 PRINTHEX PROC

159 00D4 BA 004Br MOV DX , OFFSET BUF

160 00D7 B4 09 MOV AH , 09H

161 00D9 CD 21 INT 21h

162 00DB C3 RET

163 00DC PRINTHEX ENDP

164 ;

165 00DC SIMPER PROC

166 00DC 3C 39 CMP AL , 39H

167 00DE 7F 05 JG MS1

168 00E0 2C 30 SUB AL , 30H

169 00E2 EB 10 90 JMP MSE

170 00E5 3C 46 MS1: CMP AL , 'F'

171 00E7 7F 07 JG MS2

172 00E9 2C 41 SUB AL , 'A'

173 00EB 04 0A ADD AL,10

174 00ED EB 05 90 JMP MSE

175 00F0 2C 61 MS2: SUB AL , 'a'

176 00F2 04 0A ADD AL,10

177

178 00F4 MSE:

179 00F4 C3 RET

180 00F5 SIMPER ENDP

181 00F5 DECPRINT PROC

182 ; Первод в машинное представление

183 00F5 BE 004Br MOV SI , OFFSET BUF

184 00F8 BB 1000 MOV BX , 4096

185 00FB C7 06 00AFr 0000 MOV DECW , 0

186 0101 B9 0004 MOV CX , 4

187 0104 CPER:

188 0104 8A 04 MOV AL , [SI]

189 0106 E8 FFD3 CALL SIMPER

190 0109 B4 00 MOV AH, 0

191 010B F7 E3 MUL BX

192 010D 8B 16 00AFr MOV DX , DECW

193 0111 03 D0 ADD DX , AX

194 0113 89 16 00AFr MOV DECW , DX

195 0117 D1 EB SHR BX , 1

196 0119 D1 EB SHR BX , 1

197 011B D1 EB SHR BX , 1

198 011D D1 EB SHR BX , 1

199 011F 46 INC SI

200 0120 E2 E2 LOOP CPER

201 ; Перевод в десятичное представление

202 0122 B9 0005 MOV CX , 5

203 0125 BB 2710 MOV BX , 10000

204

205 0128 MDEC:

206 0128 A1 00AFr MOV AX , DECW

207 012B BA 0000 MOV DX , 0

208 012E F7 F3 DIV BX

209 0130 89 16 00AFr MOV DECW , DX

210 0134 04 30 ADD AL , 30H

211 0136 8A D0 MOV DL , AL

212 0138 E8 000F CALL PUTCH

213 013B 8B C3 MOV AX, BX

214 013D BA 0000 MOV DX , 0

215 0140 BB 000A MOV BX , 10

216 0143 F7 F3 DIV BX

217 0145 8B D8 MOV BX , AX

218 0147 E2 DF LOOP MDEC

219 0149 C3 RET

220 014A DECPRINT ENDP

221

222 ; Процедура вывода символа на DL

223 014A PUTCH PROC

224 014A B4 02 MOV AH , 2

225 014C CD 21 INT 21H

226 014E C3 RET

227 014F PUTCH ENDP

228 ; Процедура перевода строки

229 014F LFCR PROC

230 014F B2 0A MOV DL, 10

231 0151 E8 FFF6 CALL PUTCH

232 0154 B2 0D MOV DL, 13

233 0156 E8 FFF1 CALL PUTCH

234 0159 C3 RET

235 015A LFCR ENDP

236

237 ; Procedyra vivoda simvola

238 015A GETSIMB PROC

239 ; enter simbol

240 015A B4 08 MOV AH, 08H

241 015C CD 21 INT 21H

242 015E C3 RET

243 015F GETSIMB ENDP

244 ;

245 015F MYCODE ENDS

246

247 END START

Symbol Table

Symbol Name Type Value Cref (defined at #)

??DATE Text "05/02/24"

??FILENAME Text "laba7 "

??TIME Text "22:22:46"

??VERSION Number 030A

@CPU Text 0101H

@CURSEG Text MYCODE #2 #22 #26

@FILENAME Text LABA7

@WORDSIZE Text 2 #2 #22 #26

BUF Byte DATASG:004B #15 79 159 183

CPER Near MYCODE:0104 #187 200

DECPRINT Near MYCODE:00F5 60 #181

DECW Word DATASG:00AF #16 185 192 194 206 209

ERROR Near MYCODE:00B0 #134

GETSIMB Near MYCODE:015A 84 #238

HEXADR Near MYCODE:004B 45 #76

LFCR Near MYCODE:014F 40 62 137 142 #229

MBUF Near MYCODE:00C6 90 92 94 96 98 100 102 104 106 108 110 112 114 116 118 +

120 122 124 126 128 130 132 #146

MC1 Near MYCODE:0058 #87

MCICL Near MYCODE:0051 #83

MDEC Near MYCODE:0128 #205 218

MEND Near MYCODE:003F #65 86

METLOOP Near MYCODE:0018 #43 63

MS1 Near MYCODE:00E5 167 #170

MS2 Near MYCODE:00F0 171 #175

MSE Near MYCODE:00F4 169 174 #178

MSG Byte DATASG:0010 #6 38

MSGERR Byte DATASG:00B1 #17 139

MVVOD Near MYCODE:0051 #82 152

PRINTHEX Near MYCODE:00D4 54 #158

PROC\_START Near MYCODE:004B #78 143

PUTCH Near MYCODE:014A 48 50 52 56 58 136 151 212 #223 231 233

SIMPER Near MYCODE:00DC #165 189

START Near MYCODE:0000 #28 247

TABLHEX Byte DATASG:0000 #3

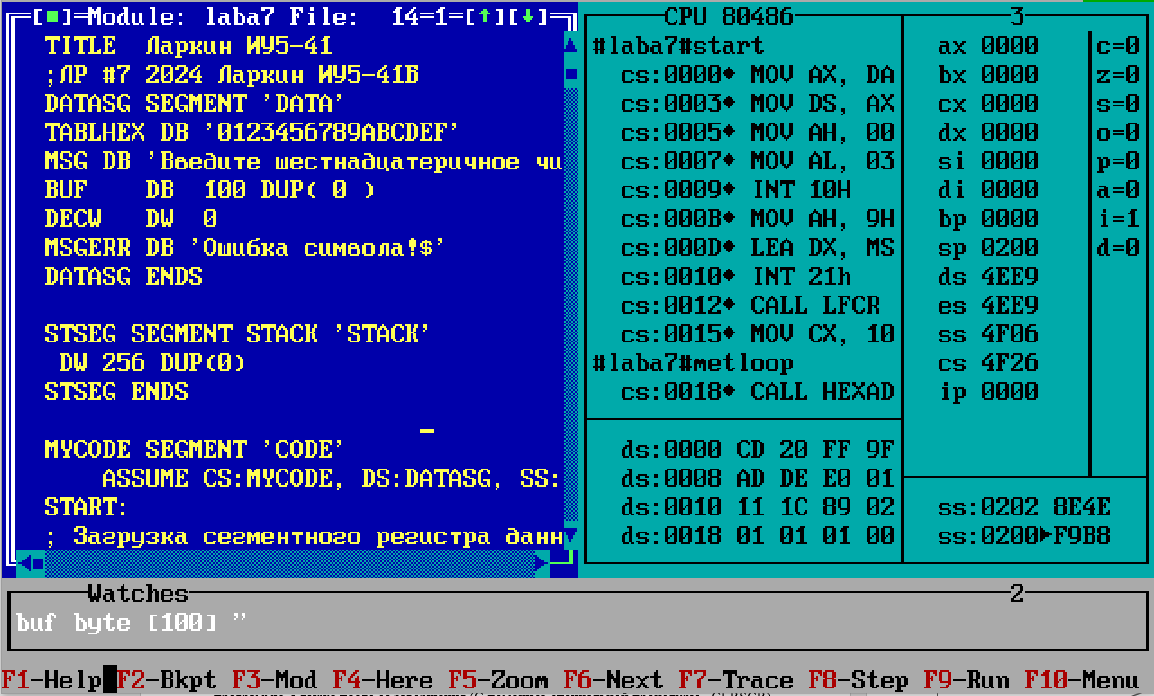
Groups & Segments Bit Size Align Combine Class Cref (defined at #)

DATASG 16 00C1 Para none DATA #2 27 30

MYCODE 16 015F Para none CODE #26 27

STSEG 16 0200 Para Stack STACK #22 27

# ­­­­6. Скриншот программы в TD.exe



# 7. Результаты работы программы

Введите шестнадцатеричное число(HHHH, \* - конец программы):

2A1C = 2A1C 10780

8D2A = 8D2A 36138

1N

Ошибка символа!

L

Ошибка символа!

92B7 = 92B7 37559

\*

# 8. Выводы по ЛР № 7

Разработан файл .ASM и соответствующие файлы приложения и листинга на языке Ассемблер. Программа выполняется в циклическом режиме до ввода ‘\*’, выводя по каждой введенной последовательности из 4х символов, принадлежащих 16й системе счисления, ее десятеричное представление. При вводе неверного символа программа выдает сообщение об ошибке и запрашивает повторный ввод.