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)

10

(количество листов)

Вариант № 11

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

Москва, МГТУ - 2024

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

**СОДЕРЖАНИЕ**

[1. Цель выполнения лабораторной работы № 7 3](#_Toc165617264)

[2. Порядок и условия проведения работы № 7 3](#_Toc165617265)

[3. Описание ошибок, возникших при отладке ЛР № 7 3](#_Toc165617266)

[4. Блок-схема программы 4](#_Toc165617267)

[5. Текст программы на языке Ассемблера (.LST) 4](#_Toc165617268)

[6. Скриншот программы в TD.exe 10](#_Toc165617269)

[7. Результаты работы программы 10](#_Toc165617270)

[8. Выводы по ЛР № 7 10](#_Toc165617271)

# 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:

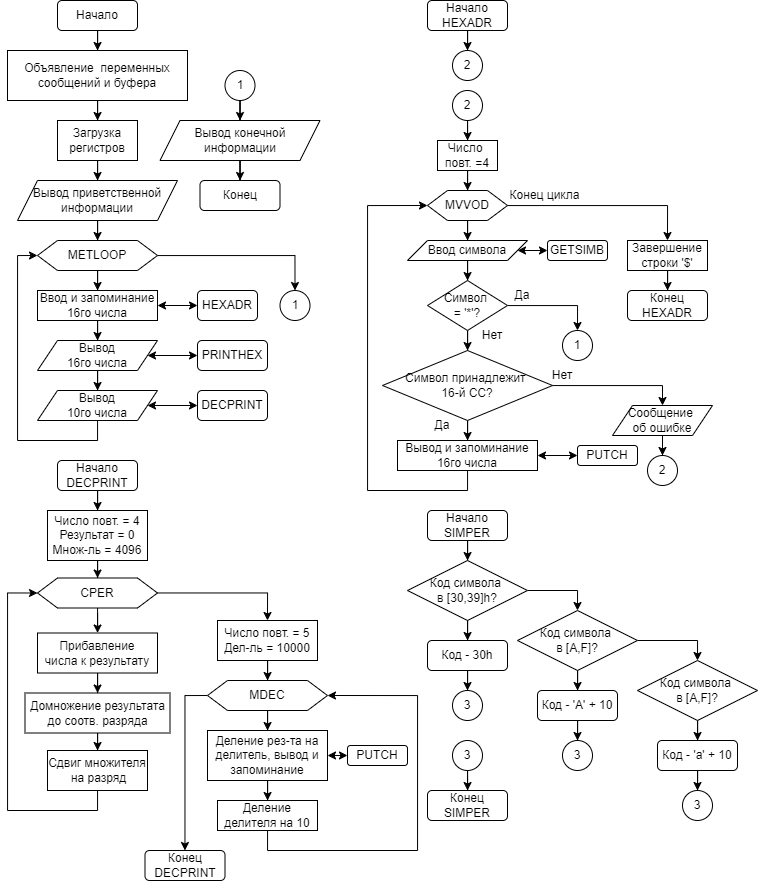
”Введите число( длинный адрес: НННН:НННН)>”:

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 05/03/24 06:22:00

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 E1 E2 AD A0+

8 A4 E6 A0 E2 A5 E0 A8+

9 E7 AD AE A5 20 E7 A8+

10 E1 AB AE 28 48 48 48+

11 48 2C 20 2A 20 2D 20+

12 AA AE AD A5 E6 20 AF+

13 E0 AE A3 E0 A0 AC AC+

14 EB 29 3A 24

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

16 00B0 0000 DECW DW 0

17 00B2 8E E8 A8 A1 AA A0 20+ MSGERR DB 'Ошибка символа!$'

18 E1 A8 AC A2 AE AB A0+

19 21 24

20 00C2 87 A0 A2 A5 E0 E8 A5+ MSGEND DB 'Завершение программы ЛР7, Ларкин Б. В. ИУ5-41, Вар. 11. Нажмите любую клавишу...$'

21 AD A8 A5 20 AF E0 AE+

22 A3 E0 A0 AC AC EB 20+

23 8B 90 37 2C 20 8B A0+

24 E0 AA A8 AD 20 81 2E+

25 20 82 2E 20 88 93 35+

26 2D 34 31 2C 20 82 A0+

27 E0 2E 20 31 31 2E 20+

28 8D A0 A6 AC A8 E2 A5+

29 20 AB EE A1 E3 EE 20+

30 AA AB A0 A2 A8 E8 E3+

31 2E 2E 2E 24

32 0113 DATASG ENDS

33

34 0000 STSEG SEGMENT STACK 'STACK'

35 0000 0100\*(0000) DW 256 DUP(0)

36 0200 STSEG ENDS

37

38 0000 MYCODE SEGMENT 'CODE'

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

40 0000 START:

41 ; Загрузка сегментного регистра данных DS

42 0000 B8 0000s MOV AX, DATASG

43 0003 8E D8 MOV DS, AX

44 ;; Цикл проверки ввода \*

45 0005 B4 00 MOV AH, 00H

46 0007 B0 03 MOV AL, 03H

47 0009 CD 10 INT 10H

48 ;

49 000B B4 09 MOV AH, 9H

50 000D BA 0010r LEA DX, MSG

51 0010 CD 21 INT 21h

52 0012 E8 0144 CALL LFCR

53 ;;

54 0015 B9 000A MOV CX, 10

55 0018 METLOOP:

56 ; Ввод шетн. числа и запоминание в BUF

57 0018 E8 003A CALL HEXADR

58 ;

59 001B B2 20 MOV DL, ' '

60 001D E8 0134 CALL PUTCH

61 0020 B2 3D MOV DL, '='

62 0022 E8 012F CALL PUTCH

63 0025 B2 20 MOV DL, ' '

64 0027 E8 012A CALL PUTCH

65 ; Вывод шестнадцатеричного числа

66 002A E8 00B1 CALL PRINTHEX

67 002D B2 20 MOV DL, ' '

68 002F E8 0122 CALL PUTCH

69 0032 B2 20 MOV DL, ' '

70 0034 E8 011D CALL PUTCH

71 ; Перевод в десятичное и печать

72 0037 E8 00C5 CALL DECPRINT

73 ; Цикл вывода шетнадцатеричной строки

74 003A E8 011C CALL LFCR

75 003D E2 D9 LOOP METLOOP

76

77 003F MEND:

78 003F B4 09 MOV AH, 9H

79 0041 BA 00C2r LEA DX, MSGEND

80 0044 CD 21 INT 21h

81 0046 E8 011B CALL GETSIMB

82 ; Очистка экрана

83 0049 B4 00 MOV AH, 00H

84 004B B0 03 MOV AL, 03H

85 004D CD 10 INT 10H

86

87 ; Выход из прораммы

88 004F B4 4C MOV AH, 4Ch

89 0051 B0 00 MOV AL, 0

90 0053 CD 21 INT 21H

91 ; Продецуры программы

92 0055 HEXADR PROC

93 ; Подготовка цикла ввода

94 0055 proc\_start:

95 0055 BE 004Cr MOV SI , OFFSET BUF

96 0058 B9 0004 MOV CX , 4

97 ; Цикл до 4-х символов

98 005B MVVOD:

99 005B MCICL:

100 005B E8 0106 CALL GETSIMB

101 005E 3C 2A CMP AL, '\*'

102 0060 74 DD JE MEND

103 0062 MC1:

104 ; Проверка символа на правильность

105 0062 3C 30 CMP AL , 30H

106 0064 74 6A JE MBUF

107 0066 3C 31 CMP AL , 31H

108 0068 74 66 JE MBUF

109 006A 3C 32 CMP AL , 32H

110 006C 74 62 JE MBUF

111 006E 3C 33 CMP AL , 33H

112 0070 74 5E JE MBUF

113 0072 3C 34 CMP AL , 34H

114 0074 74 5A JE MBUF

115 0076 3C 35 CMP AL , 35H

116 0078 74 56 JE MBUF

117 007A 3C 36 CMP AL , 36H

118 007C 74 52 JE MBUF

119 007E 3C 37 CMP AL , 37H

120 0080 74 4E JE MBUF

121 0082 3C 38 CMP AL , 38H

122 0084 74 4A JE MBUF

123 0086 3C 39 CMP AL , 39H

124 0088 74 46 JE MBUF

125 008A 3C 41 CMP AL , 'A'

126 008C 74 42 JE MBUF

127 008E 3C 42 CMP AL , 'B'

128 0090 74 3E JE MBUF

129 0092 3C 43 CMP AL, 'C'

130 0094 74 3A JE MBUF

131 0096 3C 44 CMP AL , 'D'

132 0098 74 36 JE MBUF

133 009A 3C 45 CMP AL , 'E'

134 009C 74 32 JE MBUF

135 009E 3C 46 CMP AL, 'F'

136 00A0 74 2E JE MBUF

137 00A2 3C 61 CMP AL , 'a'

138 00A4 74 2A JE MBUF

139 00A6 3C 62 CMP AL , 'b'

140 00A8 74 26 JE MBUF

141 00AA 3C 63 CMP AL, 'c'

142 00AC 74 22 JE MBUF

143 00AE 3C 64 CMP AL , 'd'

144 00B0 74 1E JE MBUF

145 00B2 3C 65 CMP AL , 'e'

146 00B4 74 1A JE MBUF

147 00B6 3C 66 CMP AL, 'f'

148 00B8 74 16 JE MBUF

149

150 00BA ERROR:

151 00BA 8A D0 MOV DL, AL

152 00BC E8 0095 CALL PUTCH

153 00BF E8 0097 CALL LFCR

154 00C2 B0 23 MOV AL,'#'

155 00C4 BA 00B2r MOV DX , OFFSET MSGERR

156 00C7 B4 09 mov AH , 09H

157 00C9 CD 21 INT 21H

158 00CB E8 008B CALL LFCR

159 00CE EB 85 JMP proc\_start

160

161 ; Запись в буфер и печать

162 00D0 MBUF:

163 00D0 88 04 MOV [SI], AL

164 00D2 46 INC SI

165 ; Печать символа

166 00D3 8A D0 MOV DL, AL

167 00D5 E8 007C CALL PUTCH

168 00D8 E2 81 LOOP MVVOD

169 ;

170 00DA C6 04 24 MOV BYTE PTR [SI], '$'

171 00DD C3 RET

172 00DE HEXADR ENDP

173 ;;

174 00DE PRINTHEX PROC

175 00DE BA 004Cr MOV DX , OFFSET BUF

176 00E1 B4 09 MOV AH , 09H

177 00E3 CD 21 INT 21h

178 00E5 C3 RET

179 00E6 PRINTHEX ENDP

180 ;

181 00E6 SIMPER PROC

182 00E6 3C 39 CMP AL , 39H

183 00E8 7F 05 JG MS1

184 00EA 2C 30 SUB AL , 30H

185 00EC EB 10 90 JMP MSE

186 00EF 3C 46 MS1: CMP AL , 'F'

187 00F1 7F 07 JG MS2

188 00F3 2C 41 SUB AL , 'A'

189 00F5 04 0A ADD AL,10

190 00F7 EB 05 90 JMP MSE

191 00FA 2C 61 MS2: SUB AL , 'a'

192 00FC 04 0A ADD AL,10

193

194 00FE MSE:

195 00FE C3 RET

196 00FF SIMPER ENDP

197 00FF DECPRINT PROC

198 ; Первод в машинное представление

199 00FF BE 004Cr MOV SI , OFFSET BUF

200 0102 BB 1000 MOV BX , 4096

201 0105 C7 06 00B0r 0000 MOV DECW , 0

202 010B B9 0004 MOV CX , 4

203 010E CPER:

204 010E 8A 04 MOV AL , [SI]

205 0110 E8 FFD3 CALL SIMPER

206 0113 B4 00 MOV AH, 0

207 0115 F7 E3 MUL BX

208 0117 8B 16 00B0r MOV DX , DECW

209 011B 03 D0 ADD DX , AX

210 011D 89 16 00B0r MOV DECW , DX

211 0121 D1 EB SHR BX , 1

212 0123 D1 EB SHR BX , 1

213 0125 D1 EB SHR BX , 1

214 0127 D1 EB SHR BX , 1

215 0129 46 INC SI

216 012A E2 E2 LOOP CPER

217 ; Перевод в десятичное представление

218 012C B9 0005 MOV CX , 5

219 012F BB 2710 MOV BX , 10000

220

221 0132 MDEC:

222 0132 A1 00B0r MOV AX , DECW

223 0135 BA 0000 MOV DX , 0

224 0138 F7 F3 DIV BX

225 013A 89 16 00B0r MOV DECW , DX

226 013E 04 30 ADD AL , 30H

227 0140 8A D0 MOV DL , AL

228 0142 E8 000F CALL PUTCH

229 0145 8B C3 MOV AX, BX

230 0147 BA 0000 MOV DX , 0

231 014A BB 000A MOV BX , 10

232 014D F7 F3 DIV BX

233 014F 8B D8 MOV BX , AX

234 0151 E2 DF LOOP MDEC

235 0153 C3 RET

236 0154 DECPRINT ENDP

237

238 ; Процедура вывода символа на DL

239 0154 PUTCH PROC

240 0154 B4 02 MOV AH , 2

241 0156 CD 21 INT 21H

242 0158 C3 RET

243 0159 PUTCH ENDP

244 ; Процедура перевода строки

245 0159 LFCR PROC

246 0159 B2 0A MOV DL, 10

247 015B E8 FFF6 CALL PUTCH

248 015E B2 0D MOV DL, 13

249 0160 E8 FFF1 CALL PUTCH

250 0163 C3 RET

251 0164 LFCR ENDP

252

253 ; Procedyra vivoda simvola

254 0164 GETSIMB PROC

255 ; enter simbol

256 0164 B4 08 MOV AH, 08H

257 0166 CD 21 INT 21H

258 0168 C3 RET

259 0169 GETSIMB ENDP

260 ;

261 0169 MYCODE ENDS

262

263 END START

Symbol Table

Symbol Name Type Value Cref (defined at #)

??DATE Text "05/03/24"

??FILENAME Text "laba7 "

??TIME Text "06:22:00"

??VERSION Number 030A

@CPU Text 0101H

@CURSEG Text MYCODE #2 #34 #38

@FILENAME Text LABA7

@WORDSIZE Text 2 #2 #34 #38

BUF Byte DATASG:004C #15 95 175 199

CPER Near MYCODE:010E #203 216

DECPRINT Near MYCODE:00FF 72 #197

DECW Word DATASG:00B0 #16 201 208 210 222 225

ERROR Near MYCODE:00BA #150

GETSIMB Near MYCODE:0164 81 100 #254

HEXADR Near MYCODE:0055 57 #92

LFCR Near MYCODE:0159 52 74 153 158 #245

MBUF Near MYCODE:00D0 106 108 110 112 114 116 118 120 122 124 126 128 130 132 +

134 136 138 140 142 144 146 148 #162

MC1 Near MYCODE:0062 #103

MCICL Near MYCODE:005B #99

MDEC Near MYCODE:0132 #221 234

MEND Near MYCODE:003F #77 102

METLOOP Near MYCODE:0018 #55 75

MS1 Near MYCODE:00EF 183 #186

MS2 Near MYCODE:00FA 187 #191

MSE Near MYCODE:00FE 185 190 #194

MSG Byte DATASG:0010 #6 50

MSGEND Byte DATASG:00C2 #20 79

MSGERR Byte DATASG:00B2 #17 155

MVVOD Near MYCODE:005B #98 168

PRINTHEX Near MYCODE:00DE 66 #174

PROC\_START Near MYCODE:0055 #94 159

PUTCH Near MYCODE:0154 60 62 64 68 70 152 167 228 #239 247 249

SIMPER Near MYCODE:00E6 #181 205

START Near MYCODE:0000 #40 263

TABLHEX Byte DATASG:0000 #3

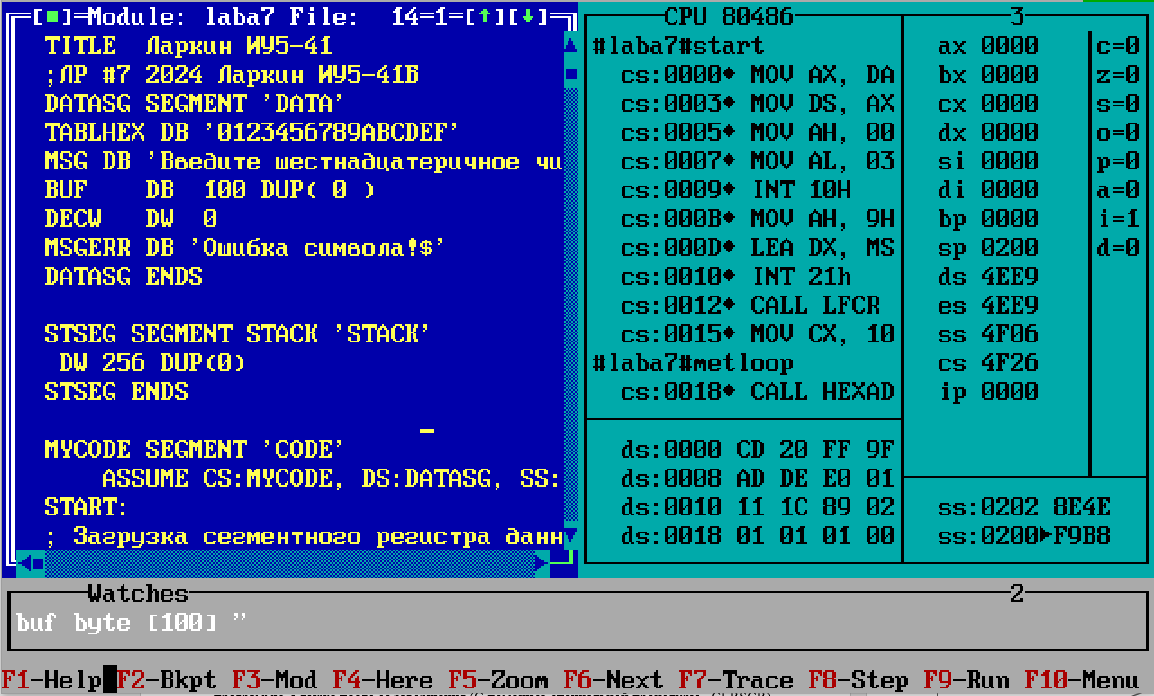
Groups & Segments Bit Size Align Combine Class Cref (defined at #)

DATASG 16 0113 Para none DATA #2 39 42

MYCODE 16 0169 Para none CODE #38 39

STSEG 16 0200 Para Stack STACK #34 39

# ­­­­6. Скриншот программы в TD.exe



# 7. Результаты работы программы

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

2A7B = 2A7B 10875

9C2F = 9C2F 39983

1892 = 1892 06290

ABCD = ABCD 43981

3N

Ошибка символа!

O

Ошибка символа!

21A3 = 21A3 08611

Завершение программы ЛР7, Ларкин Б. В. ИУ5-41, Вар. 11. Нажмите любую клавишу...

\*

# 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.