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)

15
(количество листов)
Вариант № <**3**>

ИСПОЛНИТЕЛЬ:

студент группы **ИУ5-41Б** _____
(подпись)

Бирюкова Е.И.

"__"_____2024 г.

Москва, МГТУ - 2024

СОДЕРЖАНИЕ

1. Цель выполнения лабораторной 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 – 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).

2. Порядок и условия проведения работы № 7

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

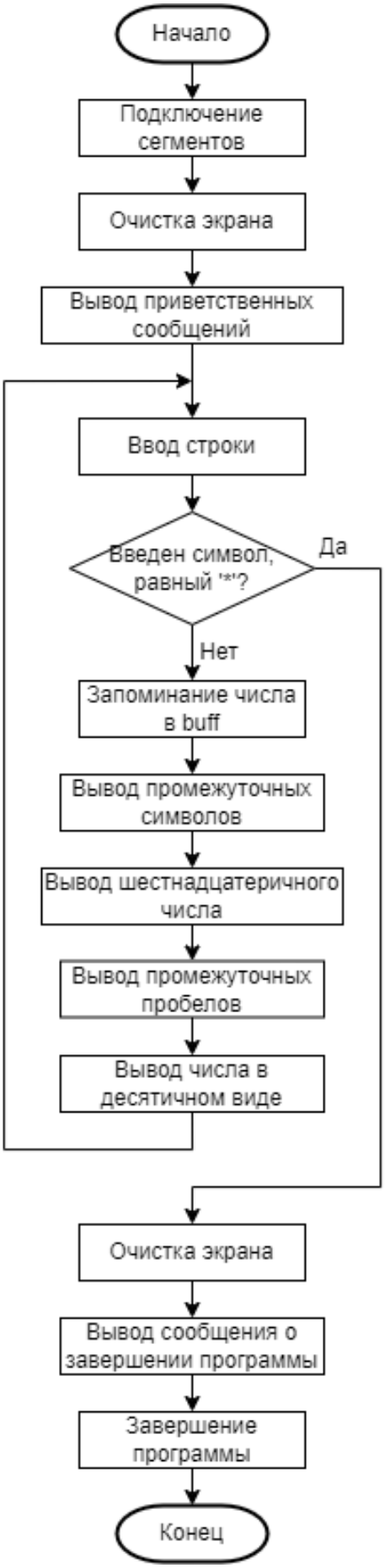
Оформить отчет по ЛР. Для оформления отчета студент должен знать или найти способ для вывода результата работы программы в текстовый файл. Лучше использовать копирование текста из окна командной строки (нежелательно снимать графическую картинку с экрана). Программа может быть выполнена в виде *.EXE исполнимого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

Вывод информации нужно выполнить с помощью функции вывода строки 09h – 021h (предварительно нужно записать введенные и выводимые данные в буферные массивы). Не забудьте в конце строки выполнить перевод строки и возврат каретки с помощью закодированных в конце строки символов 0Ah и 0Dh. Строка в каждом из массивов должна при этом завершаться символом – “$”.

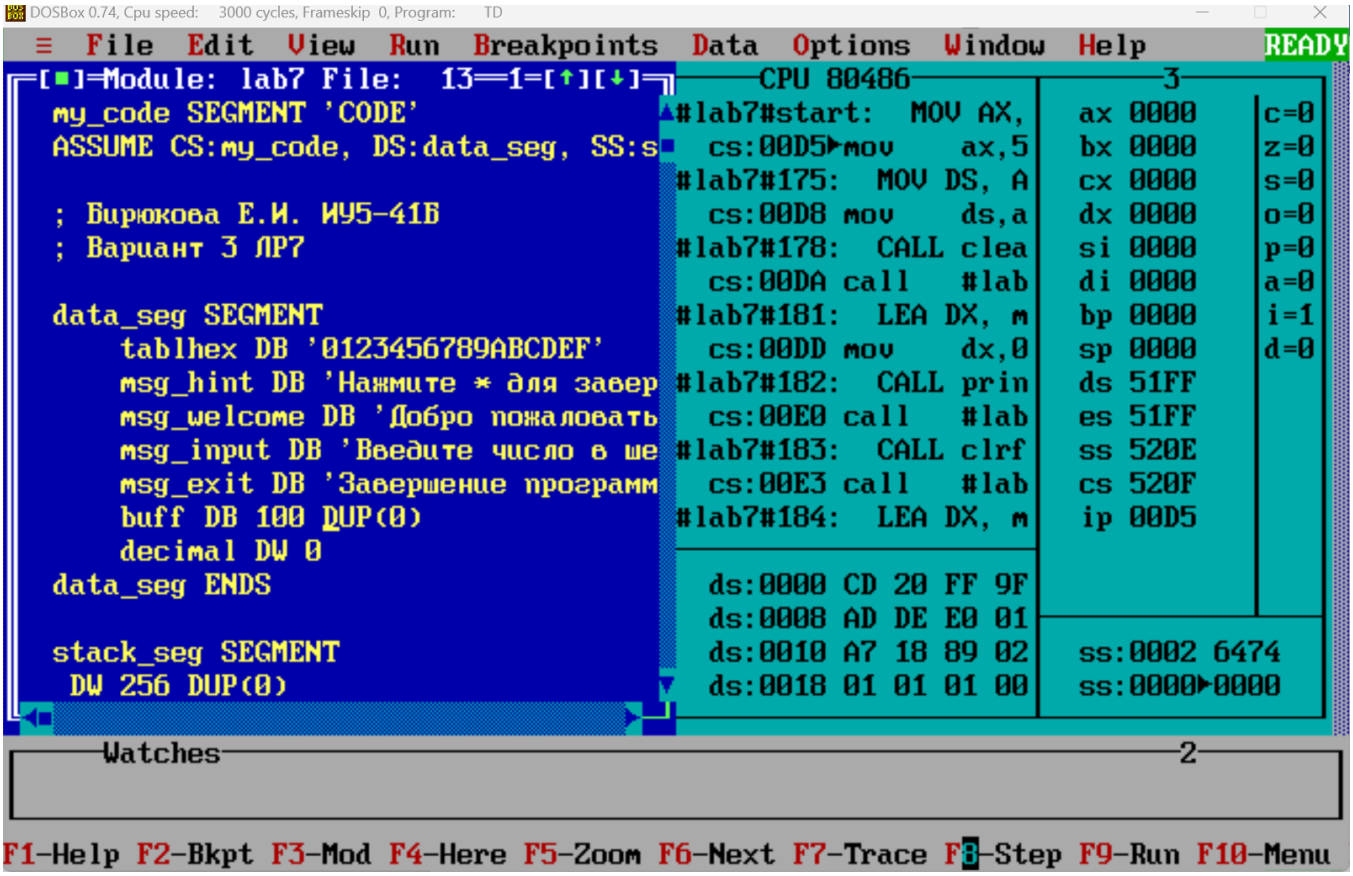
3. Описание ошибок, возникших при отладке № 7

№ п/п	Проявление ошибки	Причина ошибки	Способ устранения
1.	В процедуре hex_adr использовался переход к метке finish, если символ равен ‘*’. Ошибка при компиляции программы в этом месте.	Нельзя переходить к локальной (не глобальной) метке из одной процедуры в другую.	Прописать отдельную процедуру, которая будет содержать инструкции, которые были прописаны после метки finish. Использовать ее в обеих процедурах.
2.	При печати buff в функции print_hex программа выводила некорректные значения.	Проблема заключалась в том, что после заполнения buff не был поставлен символ конца строки.	При окончании заполнения buff допишем строку: MOV BYTE PTR [SI], '$'

4. Блок-схема программы



5. Скриншот программы в TD.exe



6. Текст программы на языке Ассемблера

Листинг программы:

Turbo Assembler Version 3.1 04/26/24 11:20:55 Page 1
lab7.asm

```
   1    0000                    my_code SEGMENT 'CODE'
   2                            ASSUME CS:my_code,      DS:data_seg, SS:stack_seg
   3
   4                            ;Бирюкова Е.И. ИУ5-41Б
   5                            ;Вариант 3 ЛР7
   6
   7    0000                    data_seg SEGMENT
   8    0000  30 31 32 33 34 35        36+     tablhex DB '0123456789ABCDEF'
   9          37 38 39 41 42 43    44+
  10          45 46
  11    0010  8D A0 A6 AC A8 E2        A5+     msg_hint DB 'Нажмите * для завершения программы.$'
  12          20 2A 20 A4 AB EF        20+
  13          A7 A0 A2 A5 E0 E8        A5+
  14          AD A8 EF 20 AF E0        AE+
  15          A3 E0 A0 AC AC EB        2E+
  16          24
  17    0034  84 AE A1 E0 AE 20        AF+     msg_welcome DB      'Добро пожаловать!!$'
  18          AE A6 A0 AB AE A2        A0+
  19          E2 EC 21 24
  20    0046  82 A2 A5 A4 A8 E2        A5+     msg_input DB 'Введите число в шестнадцатеричном
формате(HHHH):$'
  21          20 E7 A8 E1 AB AE        20+
  22          A2 20 E8 A5 E1 E2 AD+
  23          A0 A4 E6 A0 E2 A5        E0+
```

```
24        A8 E7 AD AE AC 20         E4+
25        AE E0 AC A0 E2 A5         28+
26        48 48 48 48 29 3A   24
27   0077  87 A0 A2 A5 E0 E8         A5+      msg_exit DB 'Завершение программы. Всего доброго!$'
28        AD A8 A5 20 AF E0         AE+
29        A3 E0 A0 AC AC EB         2E+
30        20 82 E1 A5 A3 AE20+
31        A4 AE A1 E0 AE A3         AE+
32        21 24
33   009C  64*(00)                   buff DB 100 DUP(0)
34   0100  0000                     decimal DW 0
35   0102                       data_seg ENDS
36
37   0000                       stack_seg SEGMENT
38   0000  0100*(0000)           DW 256 DUP(0)
39   0200                       stack_seg ENDS
40
41                              ;Вывод одного символа
42   0000                       putch PROC
43   0000  B4 02                     MOV AH, 02h
44   0002  CD 21                     INT 21h
45   0004  C3                        RET
46   0005                       putch ENDP
47
48                              ;Вывод строки
49   0005                       print_str proc
50   0005  B4 09                     MOV AH, 09h
51   0007  CD 21                     INT 21h
52   0009  C3                        RET
53   000A                       print_str endp
54
55                              ;Очистка экрана при завершении работы
56   000A                       clear_screen PROC
57   000A  B4 00                     MOV AH, 00H
```



```
58   000C  B0 03                     MOV AL, 03H
59   000E  CD 10                     INT  10H
60   0010  C3                        RET
61   0011                       clear_screen ENDP
62
63                              ;Перевод строки и возврат каретки
64   0011                       clrf PROC
65   0011  B2 0A                     MOV DL, 0AH
66   0013  E8 FFEA                   CALL putch
67   0016  B2 0D                     MOV DL, 0DH
68   0018  E8 FFE5                   CALL putch
69   001B  C3                        RET
70   001C                       clrf ENDP
71
72                              ;Вывод шестнадцатеричного числа и запись в buff
73   001C                       hex_adr PROC
74                                   ;Подготовка цикла ввода
75   001C  BE 009Cr                       MOV SI        , OFFSET buff
76   001F  B9 0004                   MOV CX        , 4
77   0022                            MVVOD:
```

```
 78   0022                           MCICL:
 79   0022  83 F9 04                 CMP CX , 4
 80   0025  74 0A                    JE    MC1
 81   0027  E8 0033                  CALL get_simb
 82   002A  3C 2A                    CMP AL, '*'
 83   002C  75 03                    JNE MC1
 84   002E  E8 0092                  CALL exit_prog
 85   0031                           MC1:
 86                                  ;Проверка символа на правильность
 87   0031  3C 30                    CMP AL , 30H
 88   0033  7C ED                    JL    MCICL
 89   0035  3C 39                    CMP AL , 39H
 90   0037  7E 08                    JLE MBUF
 91   0039  3C 41                    CMP AL , 65
 92   003B  7C E5                    JL    MCICL
 93   003D  3C 46                    CMP AL , 70
 94   003F  7F E1                    JG    MCICL
 95                                  ;Запись в буфер и печать
 96   0041                           MBUF:
 97   0041  88 04                    MOV [SI], AL
 98   0043  46                       INC SI
 99                                  ;Печать символа
100   0044  8A D0                    MOV DL, AL
101   0046  E8 FFB7                  CALL putch
102   0049  E2 D7                 LOOP MVVOD
103   004B  C6 04 24              MOV  BYTE PTR [SI], '$'
104   004E  C3                    RET
105   004F                     hex_adr ENDP
106
107                              ;Печать числа в шестнадцатеричном виде
108   004F                       print_hex PROC
109   004F  BA 009Cr                     MOV DX , OFFSET buff
110   0052  B4 09                  MOV AH , 09H
111   0054  CD 21                  INT 21h
112
113   0056  BA 0068                MOV DX, 'h'
114   0059  E8 FFA4                CALL          putch

Turbo Assembler          Version 3.1          04/26/24 11:20:55          Page 3
lab7.asm


115   005C  C3                          RET
116   005D                       print_hex ENDP
117
118                              ;Процедура ввода символа
119   005D                       get_simb PROC
120   005D  B4 08                      MOV AH, 08H
121   005F  CD 21                      INT 21H
122   0061  C3                         RET
123   0062                       get_simb ENDP
124
125   0062                       simp PROC
126   0062  3C 39                     CMP AL        , 39H
127   0064  7F 05                     JG MS1
128   0066  2C 30                     SUB AL        , 30H
129   0068  EB 03 90                  JMP MS2
130   006B                            MS1:
131   006B  2C 37                     SUB AL        , 55
132   006D                            MS2:
```

```
  133  006D  C3                  RET
  134  006E                  simp ENDP
  135
  136                        ;Вывод числа в десятичном виде
  137  006E                  print_decimal PROC
  138                        ;Перевод числа в машинное представление
  139  006E  BE 009Cr                MOV SI          , OFFSET buff
  140  0071  BB 1000             MOV BX        , 4096
  141  0074  C7 06 0100r 0000        MOV decimal, 0
  142  007A  B9 0004             MOV CX        , 4
  143  007D                      CPER:
  144  007D  8A 04                 MOV AL, [SI]
  145  007F  E8 FFE0               CALL simp
  146  0082  B4 00                 MOV AH, 0
  147  0084  F7 E3                 MUL BX
  148  0086  8B 16 0100r           MOV DX, decimal
  149  008A  03 D0                 ADD DX, AX
  150  008C  89 16 0100r           MOV decimal, DX
  151  0090  D1 EB                 SHR BX, 1
  152  0092  D1 EB                 SHR BX, 1
  153  0094  D1 EB                 SHR BX, 1
  154  0096  D1 EB                 SHR BX, 1
  155  0098  46                    INC SI
  156  0099  E2 E2               LOOP CPER
  157                        ;Перевод числа в десятичное представление
  158  009B  B9 0005             MOV CX        , 5
  159  009E  BB 2710             MOV BX        , 10000
  160  00A1                      MDEC:
  161  00A1  A1 0100r                MOV AX, decimal
  162  00A4  BA 0000               MOV DX, 0
  163  00A7  F7 F3                 DIV BX
  164  00A9  89 16 0100r           MOV decimal, DX
  165  00AD  04 30                 ADD AL, 30H
  166  00AF  8A D0                 MOV DL, AL
  167  00B1  E8 FF4C               CALL putch
  168  00B4  8B C3                 MOV AX, BX
  169  00B6  BA 0000               MOV DX, 0
  170  00B9  BB 000A                 MOV BX, 10
  171  00BC  F7 F3                 DIV BX
```



```
  172  00BE  8B D8                 MOV BX, AX
  173  00C0  E2 DF               LOOP MDEC
  174  00C2  C3                  RET
  175  00C3                  print_decimal ENDP
  176
  177                        ;Выход из программы
  178  00C3                  exit_prog PROC
  179                        ;Очистка экрана -
  180  00C3  E8 FF44             CALL clear_screen
  181
  182                        ;Сообщение о выходе из программы
  183  00C6  BA 0077r                LEA DX, msg_exit
  184  00C9  E8 FF39             CALL print_str
  185  00CC  E8 FF42             CALL clrf
  186
  187  00CF  B0 00               mov al, 00
```

```
 188  00D1  B4 4C                    mov ah, 4ch
 189  00D3  CD 21                    int 021h
 190  00D5                    exit_prog ENDP
 191
 192  00D5                    START:
 193                                 ;Подключение сегментов
 194  00D5  B8 0000s                      MOV AX, data_seg
 195  00D8  8E D8                    MOV DS, AX
 196
 197                                 ;Очистка экрана -
 198  00DA  E8 FF2D                       CALL clear_screen
 199
 200                                 ;Приветствие пользователя и запрос ввода числа
 201  00DD  BA 0034r                      LEA DX, msg_welcome
 202  00E0  E8 FF22                  CALL print_str
 203  00E3  E8 FF2B                  CALL clrf
 204  00E6  BA 0010r                      LEA DX, msg_hint
 205  00E9  E8 FF19                  CALL print_str
 206  00EC  E8 FF22                  CALL clrf
 207  00EF  BA 0046r                      LEA DX, msg_input
 208  00F2  E8 FF10                  CALL print_str
 209  00F5  E8 FF19                  CALL clrf
 210
 211  00F8  B9 0003                  MOV CX, 3
 212  00FB                           loop_main:
 213                                   ;Ввод строки
 214  00FB  E8 FF5F                    CALL get_simb
 215  00FE  3C 2A                      CMP AL, '*'
 216  0100  74 27                      JE    MEND
 217                                   ;Вывод шестнадцатеричного числа и запись в buff
 218  0102  E8 FF17                    CALL hex_adr
 219
 220                                   ;Вывод промежуточных символов
 221  0105  B2 20                      MOV DL, ' '
 222  0107  E8 FEF6                    CALL putch
 223  010A  B2 3D                      MOV DL, '='
 224  010C  E8 FEF1                    CALL putch
 225  010F  B2 20                      MOV DL, ' '
 226  0111  E8 FEEC                    CALL putch
 227
 228                                   ;Вывод шестнадцатеричного числа
```



```
 229  0114  E8 FF38                    CALL print_hex
 230
 231                                   ;Вывод промежуточного пробела
 232  0117  B2 20                      MOV DL, ' '
 233  0119  E8 FEE4                    CALL putch
 234  011C  B2 20                      MOV DL, ' '
 235  011E  E8 FEDF                    CALL putch
 236
 237                                   ;Перевод в десятичное и печать
 238  0121  E8 FF4A                    CALL print_decimal
 239  0124  E8 FEEA                    CALL clrf
 240  0127  E2 D2                    LOOP loop_main
 241
 242  0129                           MEND:
```

```
  243  0129  E8 FF97                  CALL exit_prog
  244  012C                       my_code ENDS
  245                             END START

Turbo Assembler      Version 3.1      04/26/24 11:20:55      Page 6
Symbol Table


Symbol Name          Type   Value                    Cref   (defined at #)

??DATE                      Text   "04/26/24"
??FILENAME           Text   "lab7      "
??TIME                      Text   "11:20:55"
??VERSION            Number 030A
@CPU                 Text   0101H
@CURSEG                     Text   MY_CODE                   #1  #7  #35    #37  #39
@FILENAME            Text   LAB7
@WORDSIZE            Text   2                  #1  #7  #35    #37  #39
BUFF                 Byte   DATA_SEG:009C             #33   75  109  139
CLEAR_SCREEN                Near   MY_CODE:000A              #56   180  198
CLRF                 Near   MY_CODE:0011             #64   185  203  206  209  239
CPER                 Near   MY_CODE:007D             #143   156
DECIMAL                     Word   DATA_SEG:0100             #34   141  148  150  161  164
EXIT_PROG            Near   MY_CODE:00C3             84  #178  243
GET_SIMB             Near   MY_CODE:005D             81  #119  214
HEX_ADR                     Near   MY_CODE:001C             #73   218
LOOP_MAIN            Near   MY_CODE:00FB             #212   240
MBUF                 Near   MY_CODE:0041             90  #96
MC1                  Near   MY_CODE:0031             80  83  #85
MCICL                Near   MY_CODE:0022             #78   88  92   94
MDEC                 Near   MY_CODE:00A1             #160   173
MEND                 Near   MY_CODE:0129             216   #242
MS1                  Near   MY_CODE:006B             127   #130
MS2                  Near   MY_CODE:006D             129   #132
MSG_EXIT             Byte   DATA_SEG:0077             #27   183
MSG_HINT             Byte   DATA_SEG:0010             #11   204
MSG_INPUT            Byte   DATA_SEG:0046             #20   207
MSG_WELCOME                 Byte   DATA_SEG:0034             #17   201
MVVOD                       Near   MY_CODE:0022             #77   102
PRINT_DECIMAL               Near   MY_CODE:006E             #137   238
PRINT_HEX            Near   MY_CODE:004F             #108   229
PRINT_STR            Near   MY_CODE:0005             #49   184  202  205  208
PUTCH                       Near   MY_CODE:0000             #42   66  68   101  114  167  222  224
226  233  235
SIMP                 Near   MY_CODE:0062             #125   145
START                Near   MY_CODE:00D5             #192   245
TABLHEX                     Byte   DATA_SEG:0000             #8

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

DATA_SEG             16  0102 Para    none             2  #7  194
MY_CODE                     16  012C Para    none     CODE     #1  2
STACK_SEG            16  0200 Para    none             2  #37
```

Текст программы:

```
my_code SEGMENT 'CODE'
ASSUME CS:my_code, DS:data_seg, SS:stack_seg

;Бирюкова Е.И. ИУ5-41Б
;Вариант 3 ЛР7
```

```
data_seg SEGMENT
   tablhex DB '0123456789ABCDEF'
   msg_hint DB 'Нажмите * для завершения программы.$'
   msg_welcome DB 'Добро пожаловать!$'
   msg_input DB 'Введите число в шестнадцатеричном формате(HHHH):$'
   msg_exit DB 'Завершение программы. Всего доброго!$'
   buff DB 100 DUP(0)
   decimal DW 0
data_seg ENDS

stack_seg SEGMENT
   DW 256 DUP(0)
stack_seg ENDS

;Вывод одного символа
putch PROC
   MOV AH, 02h
   INT 21h
   RET
putch ENDP

;Вывод строки
print_str proc
   MOV AH, 09h
   INT 21h
   RET
print_str endp

;Очистка экрана при завершении работы
clear_screen PROC
   MOV AH, 00H
   MOV AL, 03H
   INT  10H
   RET
clear_screen ENDP

;Перевод строки и возврат каретки
clrf PROC
   MOV DL, 0AH
   CALL putch
   MOV DL, 0DH
   CALL putch
   RET
clrf ENDP

;Вывод шестнадцатеричного числа и запись в buff
hex_adr PROC
   ;Подготовка цикла ввода
   MOV SI , OFFSET buff
   MOV CX , 4
   MVVOD:
      MCICL:
      CMP CX , 4
      JE MC1
      CALL get_simb
      CMP AL, '*'
      JNE MC1
      CALL exit_prog
      MC1:
      ;Проверка символа на правильность
      CMP AL , 30H
      JL MCICL
      CMP AL , 39H
      JLE MBUF
      CMP AL , 65
```

```
        JL MCICL
        CMP AL , 70
        JG MCICL
        ;Запись в буфер и печать
        MBUF:
        MOV [SI], AL
        INC SI
        ;Печать символа
        MOV DL, AL
        CALL putch
    LOOP MVVOD
    MOV  BYTE PTR [SI], '$'
    RET
hex_adr ENDP

;Печать числа в шестнадцатеричном виде
print_hex PROC
      MOV DX , OFFSET buff
      MOV AH , 09H
      INT 21h

      MOV DX, 'h'
      CALL putch
      RET
print_hex ENDP

;Процедура ввода символа
get_simb PROC
    MOV AH, 08H
    INT 21H
    RET
get_simb ENDP

simp PROC
    CMP AL , 39H
    JG MS1
    SUB AL , 30H
    JMP MS2
    MS1:
    SUB AL , 55
    MS2:
    RET
simp ENDP

;Вывод числа в десятичном виде
print_decimal PROC
    ;Перевод числа в машинное представление
    MOV SI, OFFSET buff
    MOV BX, 4096
    MOV decimal, 0
    MOV CX, 4
    CPER:
        MOV AL, [SI]
        CALL simp
        MOV AH, 0
        MUL BX
        MOV DX, decimal
        ADD DX, AX
        MOV decimal, DX
        SHR BX, 1
        SHR BX, 1
        SHR BX, 1
        SHR BX, 1
        INC SI
    LOOP CPER
    ;Перевод числа в десятичное представление
```

```
    MOV CX, 5
    MOV BX, 10000
    MDEC:
        MOV AX, decimal
        MOV DX, 0
        DIV BX
        MOV decimal, DX
        ADD AL, 30H
        MOV DL, AL
        CALL putch
        MOV AX, BX
        MOV DX, 0
        MOV BX, 10
        DIV BX
        MOV BX, AX
    LOOP MDEC
    RET
print_decimal ENDP

;Выход из программы
exit_prog PROC
    ;Очистка экрана
    CALL clear_screen

    ;Сообщение о выходе из программы
    LEA DX, msg_exit
    CALL print_str
    CALL clrf

    mov al, 00
    mov ah, 4ch
    int 021h
exit_prog ENDP

START:
    ;Подключение сегментов
    MOV AX, data_seg
    MOV DS, AX

    ;Очистка экрана
    CALL clear_screen

    ;Приветствие пользователя и запрос ввода числа
    LEA DX, msg_welcome
    CALL print_str
    CALL clrf
    LEA DX, msg_hint
    CALL print_str
    CALL clrf
    LEA DX, msg_input
    CALL print_str
    CALL clrf

    MOV CX, 3
    loop_main:
        ;Ввод строки
        CALL get_simb
        CMP AL, '*'
        JE MEND
        ;Вывод шестнадцатеричного числа и запись в buff
        CALL hex_adr

        ;Вывод промежуточных символов
        MOV DL, ' '
        CALL putch
        MOV DL, '='
```

```
        CALL putch
        MOV DL, ' '
        CALL putch

        ;Вывод шестнадцатеричного числа
        CALL print_hex

        ;Вывод промежуточного пробела
        MOV DL, ' '
        CALL putch
        MOV DL, ' '
        CALL putch

        ;Перевод в десятичное и печать
        CALL print_decimal
        CALL clrf
    LOOP loop_main

    MEND:
    CALL exit_prog
my_code ENDS
END START
```

7. Результаты работы программы

Запуск программы:

```
V:\TASM\TASM3>lab7.exe
```

Работа программы:

```
Добро пожаловать!
Нажмите * для завершения программы.
Введите число в шестнадцатеричном формате(HHHH):
0001 = 0001h  00001
0010 = 0010h  00016
000A = 000Ah  00010
001A = 001Ah  00026
B001 = B001h  45057
```

Завершение программы:

```
Завершение программы. Всего доброго!

V:\TASM\TASM3>
```

8. Выводы по ЛР № 7

Была создана программа, которая преобразует четырехзначные шестнадцатеричные числа, введенные пользователем, в десятичные числа. Программа выполняет преобразование в два этапа:

1. Преобразование в машинное шестнадцатеричное представление: Шестнадцатеричное число, введенное пользователем, преобразуется в машинное представление, которое может быть обработано компьютером.
2. Преобразование в десятичное представление: Затем машинное шестнадцатеричное представление преобразуется в десятичное число с использованием метода, называемого схемой Горнера.

Результаты преобразования выводятся на экран в шестнадцатеричном и десятичном форматах.

Данная программа является полезным инструментом для понимания преобразования чисел и практического применения ассемблера.