Московский государственный технический университет им. Н.Э.Баумана

|  |  |  |
| --- | --- | --- |
| Защищено:  Большаков С.А.  " " |  | Демонстрация ЛР:  Большаков С.А.  " " |

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

**Системное программирование**

**"Ввод, вывод и перевод адреса"**

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

9

(количество листов)

Вариант № 3

|  |  |
| --- | --- |
| ИСПОЛНИТЕЛЬ: |  |
| студент группы **ИУ5Ц-61Б** | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
|  | (подпись) |
| **Рябкин А.В.** | "03"\_\_Июня\_\_\_2021 г. |

Москва, МГТУ - 2021

Оглавление

[1. Цель выполнения лабораторной работы № 7 3](#_Toc73623267)

[2. Порядок и условия проведения работы № 7 3](#_Toc73623268)

[3. Описание ошибок, возникших при отладке № 7 3](#_Toc73623269)

[4. Блок-схема программы 4](#_Toc73623270)

[5. Скриншот программы KH № 7 в TD.exe 5](#_Toc73623271)

[6. Текст программы на языке Ассемблера 5](#_Toc73623272)

[7. Результаты работы программы 8](#_Toc73623273)

[8. Выводы по ЛР № 7 9](#_Toc73623274)

# Цель выполнения лабораторной работы № 7

Разработать и отладить программу на языке Ассемблер для ввода с клавиатуры четырехразрядного шестнадцатеричного числа – символами! (короткого адреса NEAR, доступные шестнадцатеричные цифры – 0123456789АBCDEF). Введенное значение переводиться

* 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
* 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

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. 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)>”:*

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

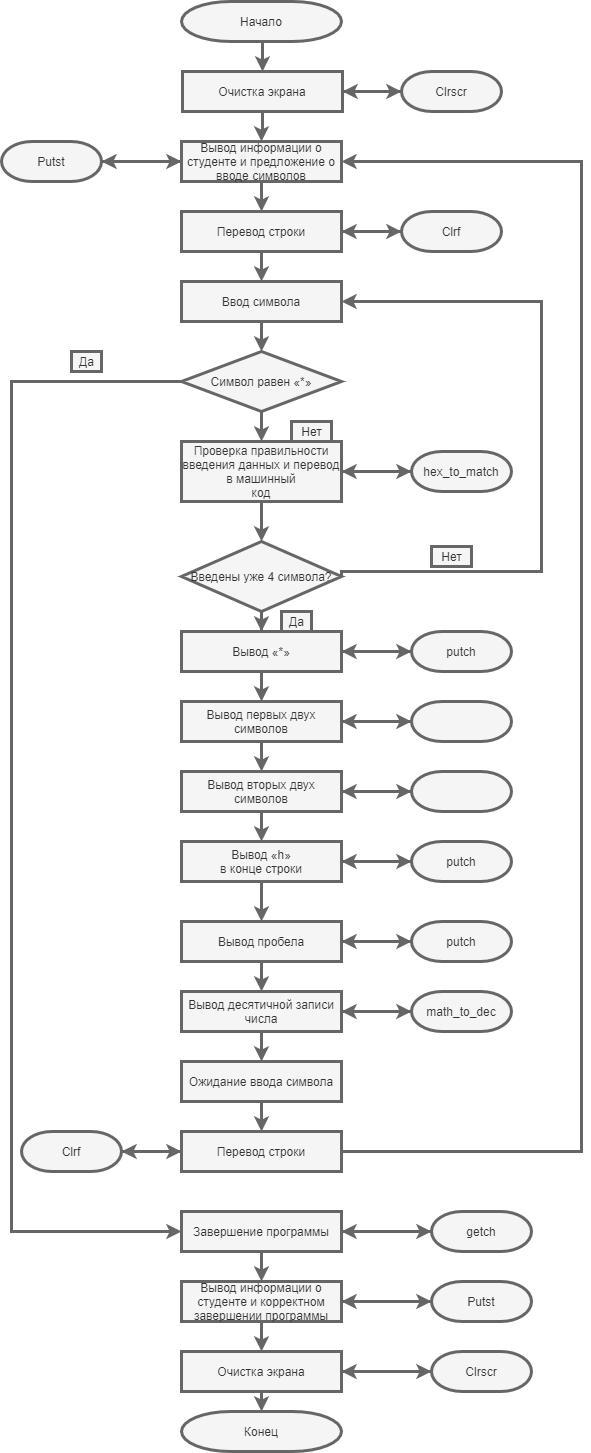
этом завершаться символом – “$”.

Необходимо оформить отчет по ЛР.

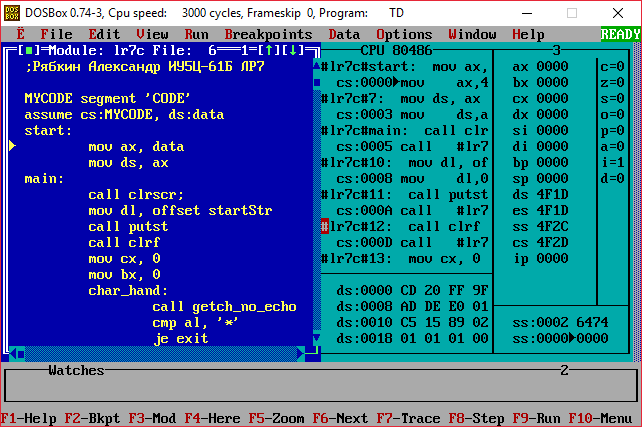
# Описание ошибок, возникших при отладке № 7

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № | Проявление ошибки | | Причина ошибки | | Способ устранения | |
| п/п |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| 1. | Некорректный | вывод | Указан | регистр | Необходимо |  |
|  | информационного | | «AX» |  | использовать | регистр |
|  | сообщения | при |  |  | «DX» |  |
|  | использовании | команды |  |  |  |  |
|  | «OFFSET» |  |  |  |  |  |
|  |  |  |  |  |  |  |
| 2. | Отсутствие | проверки | Неточное |  | Корректное |  |
|  | вводимых |  | использование | | использование | команд |
|  | шестнадцатеричных | | команд | условного | условного перехода | |
|  | символов с клавиатуры | | перехода | «JB» и |  |  |
|  |  |  | «JA» |  |  |  |
|  |  |  |  |  |  |  |

# Блок-схема программы



# Скриншот программы KH № 7 в TD.exe



# Текст программы на языке Ассемблера

;Рябкин Александр ИУ5Ц-61Б ЛР7

MYCODE segment 'CODE'

assume cs:MYCODE, ds:data

start:

mov ax, data

mov ds, ax

main:

call clrscr;

mov dl, offset startStr

call putst

call clrf

mov cx, 0

mov bx, 0

char\_hand:

call getch\_no\_echo

cmp al, '\*'

je exit

call hex\_to\_mach

cmp cx, 4

jne char\_hand

push bx

mov dl, ' '

call putch

mov dl, '='

call putch

mov dl, ' '

call putch

mov dx, bx

push dx

mov al, dh

call hex

pop dx

mov al, dl

call hex

mov dx, 'h'

call putch

mov dl, ' '

call putch

mov dl, '-'

call putch

mov dl, ' '

call putch

pop bx

call mach\_to\_dec

call clrf

call getch\_no\_echo

jmp main

exit:

call clrscr;

mov dl, offset hexStr

call putst

call clrf

call getch\_no\_echo

call clrscr;

mov al, 0

mov ah, 4ch

int 021h

putst proc

mov ah, 09h

int 021h

ret

putst endp

putch proc

mov ah, 02h

int 021h

ret

putch endp

clrf proc

mov dl, 10

call putch

mov dl, 13

call putch

ret

clrf endp

getch\_echo proc

mov ah, 01h

int 021h

ret

getch\_echo endp

getch\_no\_echo proc

mov ah, 08h

int 021h

ret

getch\_no\_echo endp

clrscr proc

mov ah, 00h

mov al, 02

int 10h

ret

clrscr endp

hex proc

mov bx, offset hexStr

push ax

shr al, 4

xlat

mov dl, al

call putch

pop ax

and al, 00001111b

xlat

mov dl, al

call putch

ret

hex endp

hex\_to\_mach proc

mov dl, al

check\_num:

cmp al,'0'

jb check\_big\_let

cmp al,'9'

ja check\_big\_let

sub al,'0'

push ax

jmp correct

check\_big\_let:

cmp al,'A'

jb check\_lit\_let

cmp al,'F'

ja check\_lit\_let

sub al,'A'

add al, 10

push ax

jmp correct

check\_lit\_let:

cmp al,'a'

jb finish

cmp al,'f'

ja finish

sub al,'a'

add al, 10

push ax

jmp correct

correct:

call putch

pop ax

mov ah, 0

shl bx, 4

add bx, ax

inc cx

finish:

ret

hex\_to\_mach endp

mach\_to\_dec proc

mov ax, bx

mov si, 0

cycle:

mov dx, 0

mov bx, mas[si]

div bx

push dx

add ax,'0'

mov dl,al

call putch

pop ax

inc si

inc si

cmp si, 10

jb cycle

ret

mach\_to\_dec endp

MYCODE ends

data segment

startStr db, 'Enter number (HHHH) :$'

hexStr db '0123456789abcdef'

mas dw 10000, 1000, 100, 10, 1

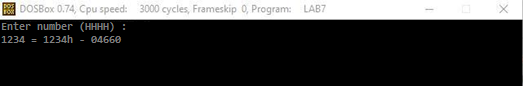
data ends

end start

# Результаты работы программы

Для проверки работоспособности программы введем число в шестнадцатеричном виде «1AB2», что в десятичной записи имеет вид «6834»

Теперь введем число «FF00» и в результате получим десятичное представление «65280»

Теперь введем число «1234», десятичное представление которого - «4660»

# Выводы по ЛР № 7

В результате выполнения этой лабораторной работы я смог разработать и отладить программу на языке Ассемблер для ввода с клавиатуры четырехразрядного шестнадцатеричного числа, осуществляя при этом проверку на корректность вводимых символов. У меня получилось с помощью циклов, команды «XLAT», команд условного перехода, команд вычитания, сложения, деления и сдвига, а также благодаря изученной схеме Горнера производить программный перевод чисел из одной системы счисления в другую.