

Защищено:
Большаков С.А.

Демонстрация ЛР:
Большаков С.А.

"__" _____ 2024 г.

"__" _____ 2024 г.

**Отчет по лабораторной работе № 4 по курсу
Системное программирование**

" Цикл и перевод символов "

(есть ли дополнительные требования - НЕТ)

9

(количество листов)

Вариант № <3>

ИСПОЛНИТЕЛЬ:

студент группы **ИУ5-41Б**

Бирюкова Е.И.

(подпись)

"__" _____ 2024 г.

СОДЕРЖАНИЕ

1. Цель выполнения лабораторной работы № 4.....	3
2. Порядок и условия проведения работы № 4	3
3. Описание ошибок, возникших при отладке № 4	3
4. Блок-схема программы.....	4
5. Скриншот программы в TD.exe.....	5
6. Текст программы на языке Ассемблера	5
7. Результаты работы программы.....	8
8. Выводы по ЛР № 4.....	9

1. Цель выполнения лабораторной работы № 4

Разработать и отладить циклическую программу на языке Ассемблер для вывода на экран 20 последовательных прописных букв русского алфавита (начиная с символа "А" или другого символа, введенного с клавиатуры). Символы должны быть представлены в символьном(печатном) и шестнадцатеричном представлении (через черточку) в виде столбчатой таблицы (см. ниже). Каждая буква выводится в виде ее символьного представления и его 2-х разрядного шестнадцатеричного числа на одной строке. Например, символ – шестнадцатеричный код: А – 80h.

2. Порядок и условия проведения работы № 4

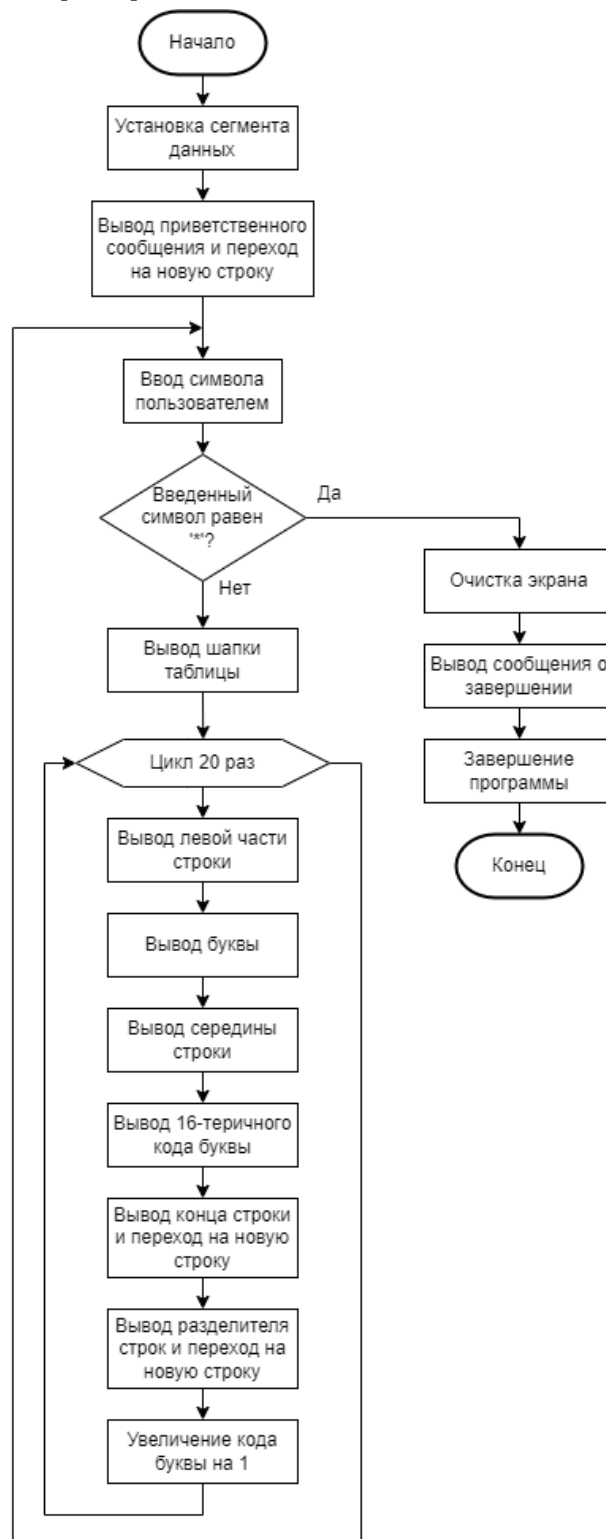
1. Начать программу с очистки экрана.
2. Создать процедуру GETCH для ввода символа без эха.
3. Создать процедуру PUTCH для вывода одного символа.
4. Создать процедуру HEX для перевода буквы в двух символьное шестнадцатеричное представление.
5. Создать процедуру CLRF для перевода строки и возврата "каретки" экрана дисплея.
6. Создать процедуру CLSSCR для очистки экрана.
7. Выполнить автоматическую шестнадцатеричную перекодировку с использованием команды XLAT и специальной таблицы перекодировки.
8. Вывести переведенные представления русских букв последовательно на экран дисплея, каждая в отдельной строке.
9. Организовать цикл с использованием команды LOOP.
10. Организовать ожидание ввода нового символа с клавиатуры с помощью процедуры GETCH.
11. Если введен символ "*", завершить программу с сообщением о завершении.
12. После завершения вывода таблицы, организовать очистку экрана.
13. Выход из программы посредством прерывания 21H - 04CH после нажатия любой клавиши, с заданием кода завершения 5.
14. Оформить отчет по лабораторной работе, вывода результат работы программы в текстовый файл без скриншотов.
15. Организовать вывод букв в виде таблицы с рамкой из одинарных линий, используя символы псевдографики.

3. Описание ошибок, возникших при отладке № 4

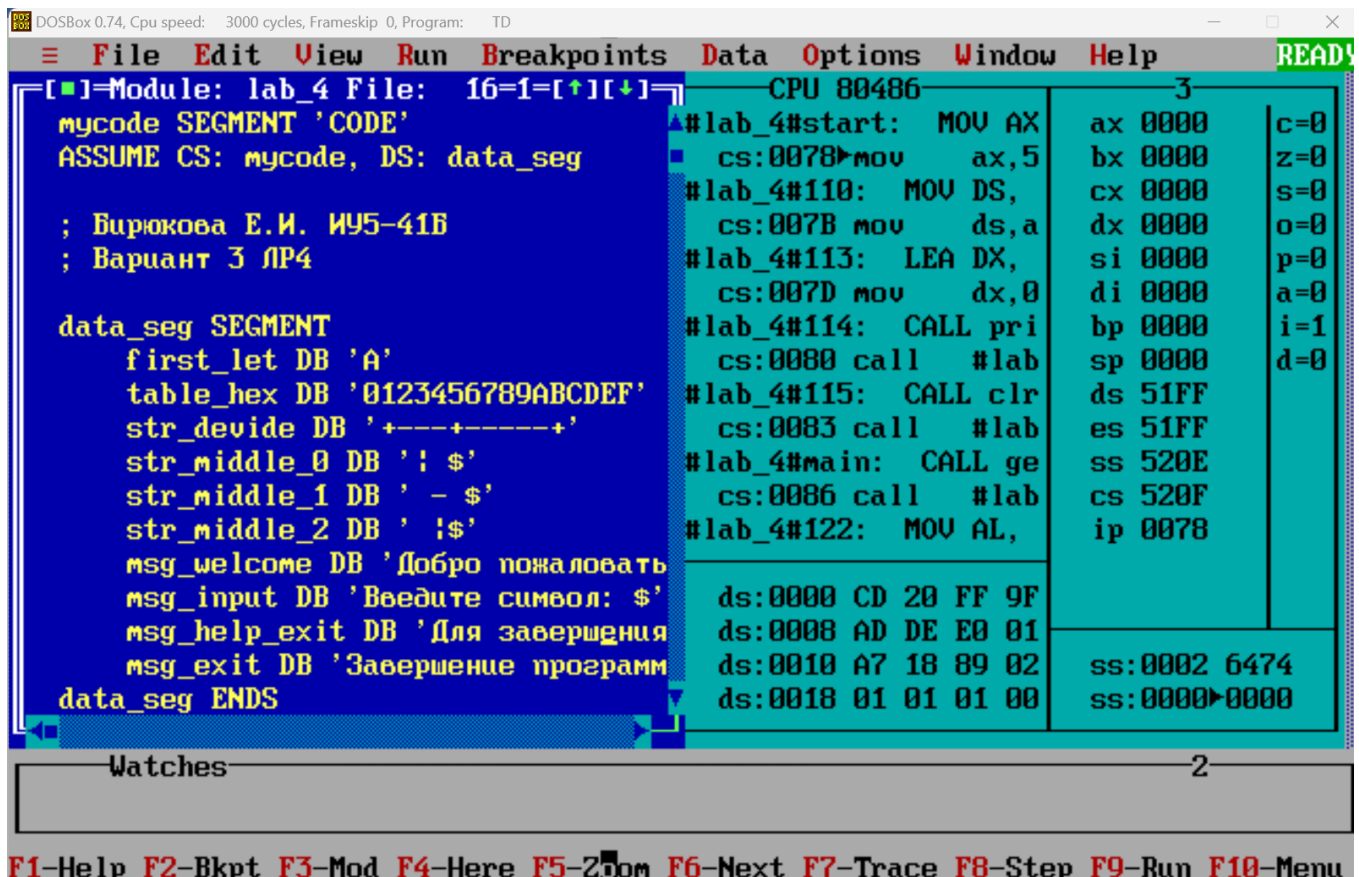
№ п/п	Проявление ошибки	Причина ошибки	Способ устранения
1.	При попытке вывести строку str_divide, которая была равна '+----+-----+', выводилась и строка str_middle_0, равная ' '. При выводе str_divide: '+----+-----+ ',	Отсутствует символ '\$ ', обозначающий конец строки.	В конец строк str_divide, str_middle_0, str_middle_1, str_middle_2, msg_welcome, msg_input, msg_help_exit и msg_exit добавлен символ '\$ '.

2.	При выводе русской буквы А соответствующий ей шестнадцатеричный код был равен 8Ah вместо 80h.	Так как first_let имела пустое значение, при записи в нее введенной буквы из регистра al менялся первый элемент строки data_hex на эту букву.	Изменение значения first_let на 'А', то есть: first_let DB 'А'.
----	---	---	---

4. Блок-схема программы



5. Скриншот программы в TD.exe



6. Текст программы на языке Ассемблера

```
mycode SEGMENT 'CODE'
ASSUME CS: mycode, DS: data_seg

; Бирюкова Е.И. ИУ5-41Б
; Вариант 3 ЛР4

data_seg SEGMENT
    first_let DB 'A'
    table_hex DB '0123456789ABCDEF'
    str_devide DB '+---+-----+'
    str_middle_0 DB '| $'
    str_middle_1 DB ' - $'
    str_middle_2 DB ' |$'
    msg_welcome DB 'Добро пожаловать!$'
    msg_input DB 'Введите символ: $'
    msg_help_exit DB 'Для завершения программы введите "*".$'
    msg_exit DB 'Завершение программы. Всего доброго!$'
data_seg ENDS

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

print_str proc
```

```

        MOV AH, 09h
        INT 21h
        RET
print_str endp

; Ввод символа
getch PROC
    ;Подсказка ввода для выхода из программы
    LEA DX, msg_help_exit
    CALL print_str
    CALL clrf

    ;Приглашение к вводу символа
    LEA DX, msg_input
    CALL print_str

    MOV AH, 01H
    INT 21H

    MOV first_let, AL
    CALL clrf

    RET
getch ENDP

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

;Очистка экрана при завершении работы
clear_screen PROC
    MOV AH, 00H
    MOV AL, 3
    INT 10H
    MOV CX, 25
    labclr:
        MOV AH, 02H
        MOV DL, 10
        INT 021H
    LOOP labclr
    RET
clear_screen ENDP

HEX PROC
    PUSH DX
    MOV AL, DL
    SHR AL, 4
    LEA BX, table_hex
    XLAT
    MOV DL, AL

```

```

    CALL putch
    POP DX
    MOV AL, DL
    AND AL, 0FH
    XLAT
    MOV DL, AL
    CALL putch
    MOV DL, 'h'
    CALL putch
    RET
HEX ENDP

```

```

exit_prog:
    ;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

```

```

START:
    MOV AX, data_seg
    MOV DS, AX

    ;Приветствие пользователя
    LEA DX, msg_welcome
    CALL print_str
    CALL clrf

main:
    ;Ввод символа
    CALL getch

    ;Проверка был ли введен символ для выхода
    MOV AL, first_let
    CMP AL, '*'

    ;Выход из программы, если введен *
    je exit_prog

    ;Вывод таблицы, если введена не *
    ;Вывод шапки таблицы
    LEA DX, str_devide
    CALL print_str
    CALL clrf

    MOV CX, 10
cycle_print:

    LEA DX, str_middle_0

```

```

CALL print_str

;Печатаем строку таблицы
MOV DL, first_let
PUSH DX
CALL putch

LEA DX, str_middle_1
CALL print_str

POP DX
CALL HEX

LEA DX, str_middle_2
CALL print_str
CALL clrf

LEA DX, str_devide
CALL print_str
CALL clrf

INC first_let

LOOP cycle_print

jmp main
mycode ENDS
END START

```

7. Результаты работы программы

Запуск программы:

```

U:\TASM\TASM3>lab_4.exe
Добро пожаловать!
Для завершения программы введите "*".
Введите символ: _

```

Ввод символа:

Для завершения программы введите "*".

Введите символ: A

```
+---+-----+
| A - 80h |
+---+-----+
| B - 81h |
+---+-----+
| B - 82h |
+---+-----+
| Г - 83h |
+---+-----+
| Д - 84h |
+---+-----+
| E - 85h |
+---+-----+
| Ж - 86h |
+---+-----+
| З - 87h |
+---+-----+
| И - 88h |
+---+-----+
| Й - 89h |
+---+-----+
```

Для завершения программы введите "*".

Введите символ: _

Завершение работы программы:

Завершение программы. Всего доброго!

U:\TASM\TASM3>_

8. Выводы по ЛР № 4

При выполнении программы были учтены следующие шаги: инициализация данных: циклический вывод букв и их шестнадцатеричных представлений в виде столбчатой таблицы с использованием псевдографики, программа завершает свою работу после вывода всех 20 букв. Результат работы программы соответствует поставленным требованиям. Таблица с символьным и шестнадцатеричным представлением русских букв алфавита была успешно выведена на экран в виде столбцов с использованием псевдографики, что обеспечивает наглядное отображение данных.