

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ
по лабораторной работе №4
по дисциплине «Организация систем и ЭВМ»
Тема «Представление и обработка символьной информации с
использованием строковых команд.»

Студент гр. 1303

Кропотов Н.Д.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

Цель работы.

Изучить представление и обработку символьной информации с использованием строковых команд на языке Ассемблера. Разработать программу, которая обрабатывает строку.

Задание.

Разработать программу обработки символьной информации, реализующую функции:

- инициализация (вывод титульной таблички с указанием вида преобразования и автора программы) - на ЯВУ;
- ввода строки символов, длиной не более N_{\max} (≤ 80), с клавиатуры в заданную область памяти - на ЯВУ; если длина строки превышает N_{\max} , остальные символы следует игнорировать;
- выполнение заданного в таблице 5 преобразования исходной строки с записью результата в выходную строку - на Ассемблере;
- вывода результирующей строки символов на экран и ее запись в файл - на ЯВУ.

Ассемблерную часть программы включить в программу на ЯВУ по принципу встраивания (in-line).

Выполнение работы.

Исключение русских букв и цифр, введенных во входной строке, при формировании выходной строки.

В начале выполнения программы в консоль выводится строка с именем, номером группы и заданием, а также просьбой ввести строку для обработки. С помощью `getline` считывается не более 81 символа с учетом символа окончания строки `'\0'`. `Setlocale` и `system` дают нам возможность работать с кириллицей.

Далее объявляется ассемблерный блок через ключевое слово `__asm`, в котором происходит считывание каждого символа введенной строки с

помощью команды `lods b`. В процессе выполнения программа проверяет каждый символ на вхождение в промежутки 'A' - 'Z', 'a' – 'z'. Если символ выходит за эти границы, то его программа игнорирует, для этого используются метки и команды перехода к меткам: `jle`, `jge`, `jmp`. Если же символ подходит под условие, то команда `stos b` записывает его в `es:edi`. Если встречается символ конца строки, ассемблерный блок заканчивается.

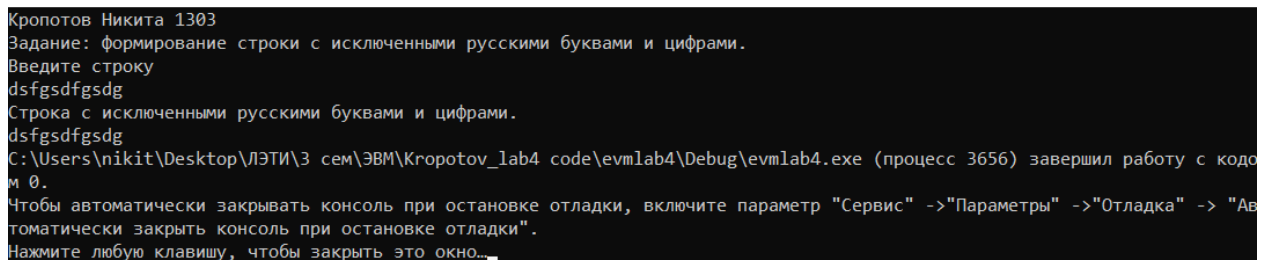
В конце, полученная строка на ЯВУ выводится на экран и записывается в текстовый файл.

Исходный код программы см. в приложении А.

Результаты тестирования программы `main.exe` представлены в табл. 1

Таблица 1 – Тестирование программы `main.exe`.

№ Теста	Ввод	Вывод	Результат
1	123 фыв вап asm sagfd erw343m	asmsagfderwm	Верно
2	dsfgsdfgsdg	dsfgsdfgsdg	Верно



```

Кропотов Никита 1303
Задание: формирование строки с исключенными русскими буквами и цифрами.
Введите строку
dsfgsdfgsdg
Строка с исключенными русскими буквами и цифрами.
dsfgsdfgsdg
C:\Users\nikit\Desktop\ЛЭТИ\3 сем\ЭВМ\Kropotov_lab4 code\evmlab4\Debug\evmlab4.exe (процесс 3656) завершил работу с кодом 0.
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" ->"Параметры" ->"Отладка" -> "Автоматически закрыть консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно.

```

Рис. 1 – Трансляция работы программы

Вывод.

В результате лабораторной работы была изучена обработка символьной информации с использованием ассемблерного блока в коде на ЯВУ.

ПРИЛОЖЕНИЕ А

Тексты исходных файлов программ main.cpp.

```
#include <iostream>
#include <fstream>
using namespace std;
char input_str[81];
char output_str[81];

int main() {
    system("chcp 1251 > nul");
    setlocale(LC_CTYPE, "rus");

    cout << "Кропотов Никита 1303\nЗадание: формирование строки с
исключенными русскими буквами и цифрами.\n";

    cout << "Введите строку\n";
    cin.getline(input_str, 81);

    ofstream file;
    file.open("result.txt");

    __asm {
        push ds
        pop es
        mov esi, offset input_str
        mov edi, offset output_str

        checking :
        lodsb
        cmp al, '\0'
        je finish
        cmp al, 'Z'
        jle checking_upper
        cmp al, 'z'
        jle checking_lower
        jmp checking
    }
```

```

        checking_upper :
    cmp al, 'A'
        jge write
        jmp checking

        checking_lower :
    cmp al, 'a'
        jge write
        jmp checking

    write :
    stosb
        jmp checking

    finish :
};

cout << "Строка с исключенными русскими буквами и цифрами.\n";
cout << output_str;
file << output_str;
file.close();

return 0;
}

```