**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 Nmax (<=80), с клавиатуры в заданную область памяти - на ЯВУ; если длина строки превышает Nmax, 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 нa 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 lodsb. В процессе выполнения программа проверяет каждый символ на вхождения 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 stosb записывает его в es:edi. Если встречается символ конца строки, ассемблерный блок заканчивается.

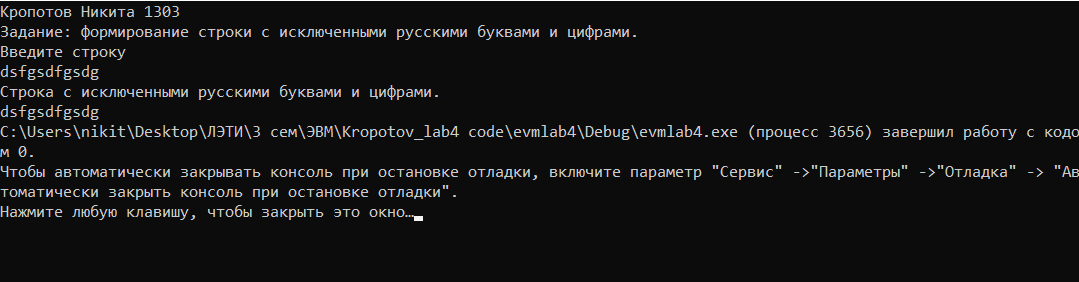
В конце, полученная строка на ЯВУ выводится на экран и записывается в текстовый файл.

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 |



*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;

}