

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

ОТЧЕТ
по лабораторной работе №4
по дисциплине «Организация ЭВМ и систем»
Тема: Представление и обработка символьной информации с
использованием строковых команд.
Вариант 14

Студент гр. 1303

Насонов Я.К.

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.

Ход работы.

Вариант 14: Исключение латинских букв и цифр, введенных во входной строке при формировании выходной строки.

Выполнение работы.

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 `lodsbyte`. В процессе выполнения программа проверяет

каждый символ через на вхождения в промежутки '0' – '9', 'A'-'Z' и 'a'-'z'. Если символ выходит за эти границы, то его программа игнорирует, для этого используются метки и команды перехода к меткам: je, jle, jl. 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 lab4.exe представлены в табл. 1.

Таблица 1 – Тестирование программы lab4.exe.

№ Теста	Ввод	Вывод	Результат
1	Пр2fgiveg53dfs5hd6т, м75ир!	Привет, мир!	Верно
2	fhYcsgsnfjhex2565	Успех	Верно
3	Teahsgстироjsf67ваниеhsfgs56	Тестирование	Верно
4	Text		Верно
5	Не нужно удалять	Не нужно удалять	Верно

Вывод.

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

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lab4.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 << "Насонов Ярослав 1303\nЗадание: Исключение латинских букв
и цифр, \nвведенных во входной строке при формировании выходной
строки\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, '0'
            jl write
            cmp al, '9'
            jle checking
            cmp al, 'A'
            jl write
            cmp al, 'Z'
            jle checking
            cmp al, 'a'
            jl write
            cmp al, 'z'
            jle checking

        write:
            stosb
            jmp checking

        finish :
    };
};
```

```
    cout << "Строка без латинских букв и цифр\n";  
    cout << output_str;  
    file << output_str;  
    file.close();  
  
    return 0;  
}
```