

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

ОТЧЕТ

по лабораторной работе №4
по дисциплине «ОргЭВМиС»

Тема: Представление и обработка символьной информации с
использованием строковых команд.

Студент гр. 0382

Афанасьев Н. С.

Преподаватели

Ефремов М.А.

Санкт-Петербург

2021

Цель работы.

Изучить представление и обработку символьной информации с использованием строковых команд на языке Ассемблера. Научиться встраивать код на языке Ассемблера в программу с кодом на ЯВУ.

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)

Вариант 2: Формирование выходной строки только из цифр и латинских букв входной строки.

Выполнение работы.

Выполнение лабораторной работы проводилось на языке C++.

В начале на ЯВУ выводится имя, номер группы и само задание. Далее считывается 81 символьный знак или меньше из потока ввода (включая терминирующий ноль) с помощью команды *fgets*.

В ассемблерном блоке, объявляемым ключевым словом `__asm`, происходит посимвольное считывание введённой строки с помощью команды *lodsb*, помещающая считавшийся из *es:esi* байт в *al*. Далее несколькими

командами *str* проверяется вхождение символа в промежуток '0-9', 'A'-'Z' и 'a'-'z'. При выходе за эти диапазоны символ игнорируется и читается следующий, иначе вызывается команда *stosb*, которая записывает байт *al* в *es:edi* – где изначально хранится отступ выходной строки. Ассемблерный код заканчивается, если встречается терминирующий ноль.

После, на ЯВУ, полученная строка выводится на экран и в текстовый файл с помощью функций стандартной библиотеки языка C++.

Разработанный программный код см. в приложении А.

Тестирование.

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Результат	Комментарии
1.	1abcaбвABCAБВ0	1abcABC0	Верно.
2.	abcABC	abcABC	Верно.
3.	12345 日本 abcdefg	12345abcdefg	Верно.

Выводы.

Изучены представление и обработка символьной информации с использованием строковых команд на языке Ассемблера. Был изучен принцип встраивания ассемблерного кода в код на ЯВУ.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЕ КОДЫ ПРОГРАММ

Название файла: lr4.cpp

```
#include <iostream>
#include <fstream>
#include <stdio.h>

char input[81];
char output[81];

int main(){
    std::cout << "Afanasyev Nazar 0382. \nvar2: input -> numbers &
latin\n";

    fgets(input, 81, stdin);
    input[strlen(input)] = '\0';

    __asm {
        push ds
        pop es
        mov esi, offset input
        mov edi, offset output
        read:
            lodsb

            cmp al, '0'
            jl next
            cmp al, '9'
            jle write

            cmp al, 'A'
            jl next
            cmp al, 'Z'
            jle write

            cmp al, 'a'
            jl next
            cmp al, 'z'
            jg next

        write:
            stosb
        next:
            cmp [esi], '\0'
            jne read
    };

    std::cout << output << std::endl;
    std::ofstream file("o.txt");
    file << output;
    return 0;
}
```