

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

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

Студентка гр. 0382

Охотникова Г.С.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2021

Цель работы.

Целью данной работы является изучение того, как работать со строковой информацией на языке Ассемблера.

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.

Вариант 12:

Формирование номера введенной латинской буквы по алфавиту и номера позиции его первого вхождения во входной строке и выдача их на экран.

Выполнение.

В начале программы инициализируются:

- `char input[81]` — статический массив элементов типа `char` для входной строки.
- `int count_alphabet` — счетчик для номера буквы в алфавите.
- `int output[26]` — статический массив целых чисел, в котором будут храниться первые вхождения букв в строку
- `int len` — переменная, в которой будет храниться длина введенной строки.

После считывания строки на языке C++ ассемблерная часть кода включается в программу по принципу in-line. В регистр `edi` загружаем смещение

на входную строку, в регистр `ecx` — длину этой строки. Команда `scasd` сканирует эту строку, а префикс `repne` осуществляет повторение этой команды, пока значение в `ecx` не станет равно нулю (то есть мы дойдем до конца строки).

На каждой итерации происходит проверка, не достигнут ли конец цепочки. Если нет, то происходит условный переход на метку для записи индекса в массив: длина строки записывается в регистр `ebx`, затем от этого регистра отнимается текущее значение регистра `ecx`. Затем переменная `count_alphabet`, которая увеличивается на единицу каждый раз, записывается в регистр `esi`, а значение, которое лежит в `ebx` записывается в массив. Переходим на метку, где проводится проверка, не является ли символ последней буквой алфавита. Если является, то происходит завершение цикла, в обратном случае, вновь перемещаемся на метку `loop` и повторяем итерацию еще раз.

Затем на языке C++ происходит вывод на экран и запись в файл. Нумерация букв в строке и в алфавите — с единицы.

Исходный программный код смотреть в приложении А.

Тестирование.

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	abcd	1 1 2 2 3 3 4 4	Программа работает корректно
2.	hello	5 2 8 1 12 3 15 5	Программа работает корректно
3.	dcba	1 4 2 3 3 2 4 1	Программа работает корректно

Выводы.

При выполнении данной лабораторной работы были изучены принципы работы со строковой информацией на языке ассемблера.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: source.cpp

```
#include <iostream>
#include <fstream>

char input[81];
int count_alphabet = -1;
int output[26] = { 0 };
int len;

int main() {
    std::cout << "Okhotnikova Galina from group 0382, lab 4" << std::endl
    << "Task: Form the number of the"
        "inputted latin letter by using the position of it both in the"
    "alphabet" << std::endl <<
        "and the relative position in the given string, then print it out"
    "on the screen" << std::endl << std::endl;

    std::cout << "Enter the line: ";
    std::cin.getline(input, 80);
    len = strlen(input);

    __asm {
        mov ax, ds
        mov es, ax
        mov esi, offset output

        mov al, 'a'
        dec al

        loop_start :
        mov edi, offset input
        mov ecx, len
        inc al
        inc count_alphabet
        repne scasb

        check :
        cmp ecx, 0
        jne write_index
        dec edi
        cmp ES : [edi] , al
        je write_index
        jmp last_latter

        write_index :
        mov ebx, len
        sub ebx, ecx
        mov esi, count_alphabet
        mov ES : output[esi * 4], ebx
        jmp last_latter

        last_latter :
        cmp al, 'z'
```

```

        je final
        jmp loop_start

    final:
};

std::ofstream file;
file.open("answer.txt");
for (int i = 0; i < 26; i++) {
    if (output[i] != 0) {
        std::cout << i + 1 << ' ' << output[i] << std::endl;
        file << i + 1 << ' ' << output[i] << std::endl;
    }
}
file.close();

return 0;
}

```