

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

Студент гр. 1381

Смирнов Д. Ю.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

Цель работы.

Изучить представление и обработку строк 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 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 12:

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

Выполнение работы.

Создаем глобальные переменные:

`char input_str[81]` - массив элементов типа `char` для входной строки.

`int letters[26] = {0}` – массив элементов тип `int`, в котором будут храниться номер первого вхождения буквы.

`int counter` – счетчик букв для `letters`.

Считывание строки на языке C++ ассемблерная часть кода включается в программу по принципу in-line. В регистр `es` записываем смещение на `ds`, а в регистр `esi` записываем смещение на строку `input_str`. В `ah` заносим символ перед прописной латинской буквой 'a'.

Метки:

- *loop_s*: В *esi* записываем смещение на строку *input_str*. Увеличиваем *ah* и *counter* на 1. Очищаем регистр *ecx*.
- *loop_*: Очищаем нижний байт регистра *ah*. Используя *lodsb* загружаем в *al* символ из строки *input_str*. Увеличиваем регистр *ecx* — количество считанных символов. Если считанный символ является нуль-терминатором переходим на метку *check_last_letter*.
- *check_letter*: Проверяем текущую букву, если она является искомой, то идем на метку *write_index*, Иначе шаг на метку *check_last_letter*.
- *write_index*: В массив *letters* по индексу *counter* (масштабируются в 4 раза, так как *letters* массив типа *int*) записывается регистр *ecx*. Шаг на метку *loop_s*.
- *check_last_letter*: Если искомый символ является 'z', то переход на метку *final*. Если символ конца строки, то переход на метку *loop_s*. иначе на *loop_*.

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

Тестирование.

Результаты тестирования представлены в табл. 1.

Таблица 1 – Тестирование программы

№	Входные данные	Результат	Комментарий
1	happy	1 2 8 1 16 3 25 5	Верный результат
2	lab LAB	1 2 2 3 12 1	Верный результат
3	badcA	1 2 2 1 3 4 4 3	Верный результат

Выводы.

Изучены представление и обработка строк на языке Ассемблера.

ПРИЛОЖЕНИЕ А

КОД ПРОГРАММЫ

Название файла: lab4.cpp

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

char input_str[81]; // Входная строка
int letters[26] = {0}; // Массив вхождений букв (если буква есть в строке
0 меняется на её индекс)
int counter = -1; // индекс массива letters

// Вариант 12. Формирование номера введенной латинской буквы по алфавиту
// и номера позиции его первого вхождения во входной строке и выдача их
на экран.

int main() {
    std::cout << "Smirnov Dmitry group - 1381\n";
    std::cout << "Task: Forming the number of the entered Latin letter in
alphabetical order\nand the position number of its first occurrence in
the input string\nand displaying them on the screen.\n";
    std::cout << "Enter string:\n";
    std::cin.getline(input_str, 81);

    __asm {
        push ds
        pop es
        sub ax, ax
        mov ah, 96 // Символ перед латинской буквой 'a'
    loop_s:
        mov esi, offset input_str // адрес начала строки
        inc ah // Ищем следующий латинский символ
        inc counter // увеличиваем индекс letters
        sub ecx, ecx // очищаем ecx
    loop_:
        sub al, al // очищаем al
        lodsb // загружаем символ из строки input_str в al
        inc ecx // увеличиваем кол-во считанных символов
        cmp al, '\0' // если строка закончилась идем на метку
check_last_letter
        je check_last_letter
    check_letter:
        cmp al, ah
        je write_index // найден прописной символ

        mov bh, ah
        sub bh, 32
        cmp al, bh
        je write_index // найден заглавный символ

        jmp check_last_letter

    write_index:
        mov edi, counter
```

```

        mov ES:letters[edi * 4], esx // записываем номер вхождения
символа (масштабируем в 4 раза тк массив типа int)
        jmp loop_s // начинаем новую итерацию цикла
check_last_letter:
        cmp ah, 'z' // если дошли до z завершаем
        je final // метка на конец вставки
        cmp al, '\0' // строка закончилась начинаем новый цикл
        je loop_s // метка на начало цикла
        jmp loop_ // продолжаем считывание строки

final:
};
std::fstream file;
file.open("./answer.txt");
for (int i = 0; i < 26; i++)
{
    if (letters[i] != 0) {
        std::cout << i + 1 << " " << letters[i] << std::endl;
        file << i + 1 << " " << letters[i] << std::endl;
    }
}
file.close();
std::cout << "Completed!\n";
return 0;
}

```