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 Nmax (<=80), с клавиатуры в заданную область памяти на ЯВУ; если длина строки превышает Nmax, 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 хранится номер первого вхождения буквы.

int counter – счетчик букв для letters.

int len – длинна входной строки.

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 массив letters.

Метки:

- loop_: В регистр edi загружается смещение на входную строку, в регистр ecx—длину этой строки. Команда scasb сканирует эту строку, а префикс repne осуществляет повторение этой команды, пока значение в ecx не станет равно нулю (то есть текущий символ является нуль-терминатором).
- check_letter: Если строка считана до конца (ecx = 0), то идем на метку write_index, иначе проверяем прошлую букву. Если прошла буква является искомой, то идем на метку write_index. Шаг на метку check_last_letter.
- write_index: Рассчитывается номер вхождения буквы, записывается в массив letters по индексу counter (масштабируются в 4 раза, так как letters массив типа int). Шаг на метку check_last_letter.
- check_last_letter: Если искомый символ является ‘z’, то переход на метку final, иначе переход на метку 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	1 2 2 3 12 1	Верный результат
3	badc	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 строчка
int letters[26] = {0}; // Массив вхождений букв (если буква есть в строке
-1 меняется на её индекс)
int counter = -1; // индекс массива letters
int len;

// 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);
    len = strlen(input_str);
    __asm {
            mov ax, ds
            mov es, ax
            mov esi, offset letters // адрес массива letters

            mov al, 96 // Символ перед латинской буквой 'a'
    loop_:
            mov edi, offset input_str // адрес начала строки
            mov ecx, len // записываем длинну строчки
            inc al // Ищем следующий латинский символ
            inc counter // увеличиваем индекс letters
            repne scasb // сканируем строку пока не найдем нуль-
терминатор

    check_letter:
            cmp ecx, 0
            jne write_index // строка считана до конца то записываем
индекс
            dec edi // проверяем прошлую букву
            cmp ES:[edi], al
            je write_index // буква = искомой, то записываем индекс
            jmp check_last_letter // провекра на последнюю букву
латинского алфавита

    write_index:
            mov ebx, len // ebx = длине строки
            sub ebx, ecx // (длина строки - ecx) = индексу вхождения
буквы
            mov esi, counter // берем индекс для записи
```

```
            mov ES:letters[esi * 4], ebx // записываем индекс вхождения
(маштабируем в 4 раза тк массив типа int)
            jmp check_last_letter // проверяем все ли буквы пройдены

    check_last_letter:
            cmp al, 'z' // если дошли до z завершаем
            je final // метка на конец вставки
            jmp loop_ // если не последяя буква латинского языка, то
заново сканируем строчку

    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;
}
```