

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. 0383

Рудкова Ю.В.

Преподаватель:

Ефремов М.А.

Санкт-Петербург

2021

Цель работы

Изучить принципы обработки символьной информации с использованием строковых команд на языке Ассемблер. Написать программу на языке высокого уровня со вставкой ассемблерного кода по принципу in-line.

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 12:

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

Ход выполнения

в ходе работы была разработана программа на языке C++ со встраиванием ассемблерного кода по принципу in-line.

Объявим, инициализируем необходимые переменные: `char input[81]` - под входную строку, `int num=-1` - для счетчика номеров букв в алфавите на каждой итерации поиска, `int arr[26]={0}` - под массив номеров позиций первого вхождения букв латинского алфавита, `int len` - для длины входной строки, понадобится в дальнейшем для `esx` - счетчика повторений выполнения команды `scasb` с префиксом `repne`.

Выведем строку инициализации работы с указанием автора и вида преобразования строки: `std::cout<<"Автор: Рудакова Юлия ст.гр.0383\nФормирование номера введенной латинской буквы по алфавиту и номера позиции его первого вхождения во входной строке и выдача их на экран.\nВведите строку:"`

Считываем входную строку `cin.getline(input, 81)`. И вычислим ее длину `len = strlen(input)`, она нам понадобится в дальнейшем для счетчика повторений `ecx` в цикле поиска символа в строке.

Преобразование будем осуществлять посредством разработки ассемблерного кода и вставки его блока по принципу `in-line: __asm{}`.

Будем полагать, что нумерация букв в алфавите начинается с единицы, также будем считать, что позиции букв в слове тоже начинаются с единицы.

Для поиска символа во входной строке в дальнейшем будем использовать строковую команду `SCASB`, которая сравнивает элемент строки, адрес которого задается парой `ES:DI`, со значением регистра `AL` и результат сравнения фиксирует в флагах, после чего `DI` увеличивается на 1. Поэтому мы должны установить `edi` на смещение `input` входной строки. Перед строковой командой `SCASB` поставим префикс повторения `REPNE`, который устанавливает `ZF` в 0 и, постоянно уменьшая `CX`, заставляет многократно повториться эту команду до момента, когда `ZX` установится 1 или `CX` не достигнет 0. В регистр `ECX` записываем значение длины строки, вычисленное ранее, `len` - именно столько раз выполнится `SCASB` - сравнение элемента строки с искомым символом.

Для поиска латинских букв во входной строке нужен цикл перебора всех букв латинского алфавита в регистре `AL` и на каждой итерации запускать поиск текущего символа во входной строке и увеличивать значение счетчика `num`, который будет отвечать за порядковый номер буквы в алфавите, поиск которой сейчас происходит. Если заданный символ найден во входной строке, то обращаемся к `arr[esi*4]`, где `esi` предварительно присваивается значение `num`, и в эту ячейку памяти будем записывать индекс первого вхождения буквы в строку (умножаем на 4, так как `arr` - целочисленный массив, каждый элемент в котором занимает 4 байта). Если заданный символ не найден, то переходим к следующей букве алфавита. После выполнения алгоритма получим массив `arr[]`, в котором каждому элементу с индексом-номером буквы в латинском алфавите соответствует число - индекс первого вхождения этой буквы во входной строке. Если массив заполнен нулями, т.е. латинских букв не встретилось, то выведем "Буквы латинского алфавита отсутствуют в введенной строке".

После каждого выполнения `SCASB` значения `ECX` и `ES:DI` будут меняться, отличаясь от первоначальных, указывающих на начало строки и длину строки. Поэтому на каждой итерации цикла нужно переопределять их вновь с помощью `move di, offset input` и `move ecx, len`.

Перебирать символы алфавита будем с помощью команды INC AL (так мы переберем все символы от а до z), опираясь на таблицу кодировки символов CP-866.

Выполняя поиск в строке символа из AL, проверяем если ECX != 0, значит, не все символы строки перебрались и где-то встретилась искомая буква. Используя команду условного перехода, перейдем к метке get_index, где перейдем к вычислению индекса первого вхождения. Если же ECX = 0, значит, все символы в строке перебрались. Но возможны два варианта: все символы перебрались и не нашли искомого, все символы перебрались и последний символ оказался искомым. Поэтому мы должны сравнить последний символ с искомым с помощью cmp. В случае эквивалентности - нашли искомый символ и переходим к вычислению индекса первого вхождения по метке get_index. В ином случае - искомый символ в строке отсутствует, можем перейти к поиску следующей буквы, но при этом проверим, если текущая буква поиска оказалась символом “z” - значит, все буквы алфавита перебраны, входная строка обработана, можем выводить результат.

Тестирование

Табл.1 - Результаты тестирования программы

№ п/п	Входные данные	Выходные данные	Комментарии
1.	abcd	1 1 2 2 3 3 4 4	Верно
2.	зюляzxsx	19 7 24 6 26 1	Верно
3.	ассемблер	Буквы латинского алфавита отсутствуют в введенной строке.	Верно

Выводы

В результате работы была изучена обработка символьной информации с использованием строковых команд на языке Ассемблер посредством разработки программы, формирующей номер введенной латинской буквы по алфавиту и номера позиции ее первого вхождения во входной строке и выводящей их на экран.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.cpp

```
#include <iostream>

using namespace std;

char input[81];
int num = -1; //счётчик-номер буквы в алфавите
int arr[26] = { 0 }; //массив номеров позиций первого вхождения буквы латинского
алфавита во входной строке
int len;

int main() {
    system("chcp 1251 > nul");
    setlocale(LC_CTYPE, "rus");
    cout << "Автор: Рудакова Юлия ст.гр.0383 \nФормирование номера введенной
латинской буквы по алфавиту и номера позиции его первого вхождения во входной
строке и выдача их на экран.\nВведите строку: ";

    cin.getline(input, 81); //входная строка
    len = strlen(input);    //длина входной строки
    __asm {
        push ds
        pop es

        mov al, 'a'
        dec al
        cycle ://перебор букв алфавита
        mov edi, offset input //устанавливаем в edi смещение на входную строку
        mov ecx, len
        cmp al, 'Z'
        je label
```

```

    inc al
    label ://поиск в строке
    inc num
    repne scasb    //посимвольный поиск во входной строке, он повторится
    такое количество раз, сколько во входной строке символов
    has_symbol_check ://проверка символа
    cmp ecx, 0
    jne get_index //если символ встретился
    dec edi
    cmp ES : [edi] , al //проверка вдруг последний символ и есть искомый
    je get_index
    jmp last_iteration_check
    get_index ://если символ встретился
    mov ebx, len
    sub ebx, ecx //ebx теперь содержит индекс первого вхождения
    mov esi, num
    mov ES : arr[esi * 4], ebx //положили в массив индекс первого вхождения
    jmp last_iteration_check
    last_iteration_check ://вызов проверки следующего символа алфавита
    cmp al, 'z'
    je the_end
    jne cycle

    the_end :
};
int flag = 0;
for (int i = 0; i < 26; i++) {
    if (arr[i] != 0) {
        cout << i + 1 << ' ' << arr[i] << endl;
        flag = 1;
    }
}
if (flag == 0)
    cout << "Буквы латинского алфавита отсутствуют в введенной строке.";

```

```
    return 0;  
}
```