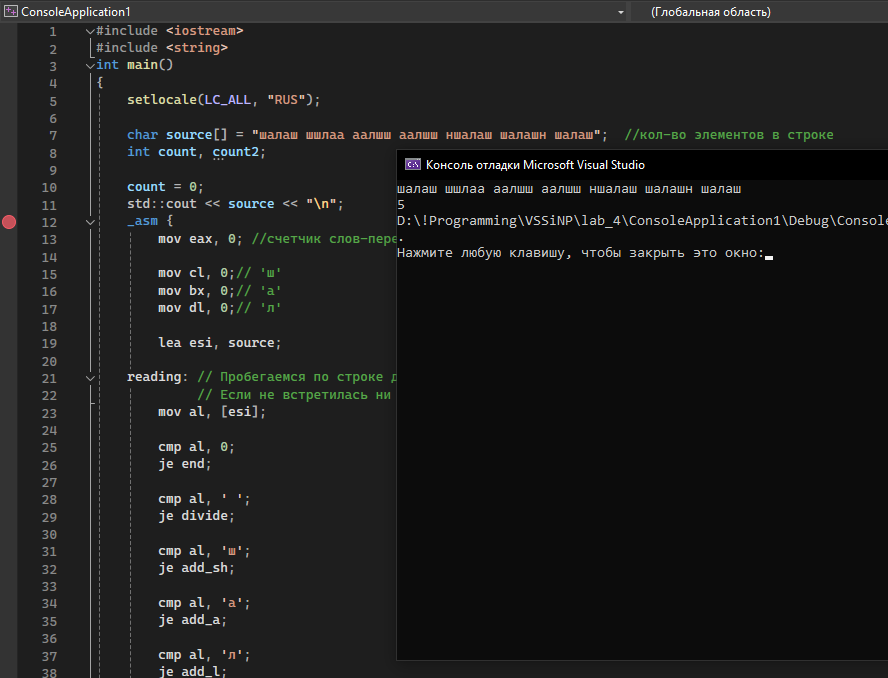
|  |  |
| --- | --- |
| МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  Федеральное государственное автономное образовательное учреждение  высшего профессионального образования  **«Дальневосточный федеральный университет»** | |
| **ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ** | |
| **Департамент программной инженерии и искусственного интеллекта** | |
| Вычислительные системы сети и низкоуровневое программирование | |
| Отчет по лабораторной работе №4  «Работа со строками с помощью ассемблерных вставок» | |
|  | |
|  | Выполнил студент гр. Б9122-09.03.04  Лопатин А.О.  Проверил преподаватель  Терентьева А.М.  \_\_\_\_\_\_\_\_\_ |
| г. Владивосток  2024 г. | |

**Цель лабораторной работы**

Целью данной работы является разработка и реализация программы для работы со строками с помощью ассемблерных вставок.

Скриншот экрана или содержимое файла вывода:



Текст программы

#include <iostream>

#include <string>

int main()

{

setlocale(LC\_ALL, "RUS");

char source[] = "шалаш шшлаа аалшш аалшш ншалаш шалашн шалаш"; //кол-во элементов в строке

int count, count2;

count = 0;

std::cout << source << "\n";

\_asm {

mov eax, 0; //счетчик слов-перевертышей слова "шалаш"

mov cl, 0;// 'ш'

mov bx, 0;// 'a'

mov dl, 0;// 'л'

lea esi, source;

reading: // Пробегаемся по строке до разделителя ' ' и до букв ш а л. Если встречается буква ш или л или а, увеличиваем cl, dl, bx соответственно.

// Если не встретилась ни одна из букв, переходим до следующего пробела

mov al, [esi];

cmp al, 0;

je end;

cmp al, ' ';

je divide;

cmp al, 'ш';

je add\_sh;

cmp al, 'а';

je add\_a;

cmp al, 'л';

je add\_l;

jmp skip;

next: //следующий символ

inc esi;

jmp reading;

skip:

inc esi;

cmp[esi], ' ';

jne skip;// пропускаем пока не встретим слово

cmp[esi], 0;

je end;

// зануляем счётчики для следующего слова

mov cl, 0;

mov bx, 0;

mov dl, 0;

jmp reading;

divide:

inc esi;

cmp cl, 2; // 'ш' = 2

je check\_a;

mov cl, 0;

mov bx, 0;

mov dl, 0;

jmp reading;

check\_a:

mov cl, 0;

cmp bx, 2;// 'a' = 2

je check\_l;

mov bx, 0;

mov dl, 0;

check\_l:

mov bx, 0;

cmp dl, 1; // 'л' = 1

je summ\_count;

mov dl, 0;

summ\_count:

mov dl, 0;

mov eax, 0; //counter

inc eax;

cmp count, 0;

jg add\_count; //count > 0

mov count, eax;

jmp reading;

add\_count:

inc count;

jmp reading;

add\_sh:

inc cl;

jmp next;

add\_a:

inc bx;

jmp next;

add\_l:

inc dl;

jmp next;

end:

cmp cl, 2;

je check\_a;

}

std::cout << count;

return 0;

}