Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №1

за 2 семестр

По дисциплине: «Компьютерные системы и сести »

Выполнил:

Студент 2 курса

Группы ПО-6(2)

Осьмушеиков А.Ю.

Проверил:

Бойко Д.О.

Брест, 2022

Лабораторная работа №1

Вариант 10

**Задание:** написать ассемблерную вставку, реализующую обработку строки

согласно варианту. Оформить ее в виде отдельной функции.

3. Реализовать данную обработку строки также в виде функции на С++.

4. Сравнить быстродействие обоих вариантов. В отчете отразить выводы.

**Текст программы:**

**Код на Ассемблере:**

#include <iostream>

#include <windows.h>

#include <string.h>

#include <tchar.h>

#include <ctime>

void MoveASM(char\* str, size\_t cnt)

{

    size\_t len = strnlen\_s(str, 32) - 1;

    \_\_asm

    {

        push eax         //сохранение eax

        push ecx         //сохранение ecx

        push ebx         //сохранение ebx

        mov ecx, cnt         //копируем cht в регистр ebx

        lbl1 :            //метка

        push ecx          //сохранение eсx

            xor eax, eax   //очищаем регистр

            mov ebx, str      //копируем str в регистр ebx

            mov al, byte ptr[ebx]

            push eax         //сохранение eax

            mov ecx, len     //копируем len в регистр ebx

            mov esi, str     //копируем str в регистр esi

            inc esi

            mov edi, str     //копируем str в регистр edi

            rep movsb

            pop eax         //извлекаем ebx из стека

            mov ebx, str    //копируем str в регистр ebx

            add ebx, len     // складываем ebx и len, затем помещаем сумму в ebx

            mov byte ptr[ebx], al      //записываем 8 бит в ebx

            pop ecx                 //извлекаем eсx из стека

            dec ecx //уменьшаем ecx на 1

            jecxz ext        //проверяем значение регистра ecx и если оно равно 0 , то передаем переходим к метке ext

            jmp lbl1       // переходим к метке lbl1

            ext :            // метка

        pop ebx            //извлекаем ebx из стека

            pop ecx        //извлекаем eсx из стека

            pop eax        //извлекаем eаx из стека

    }

}

int \_tmain(int argc, \_TCHAR\* argv[])

{

    SetConsoleCP(1251);

    SetConsoleOutputCP(1251);

    char\* str = new char[32];

    std::cout << "Введите строку:\n";

    std::cin >> str;

    srand(time(0));

    MoveASM(str, 5);

    std::cout << "\nРезультат:\n" << str;

    std::cout << "\nВремя выполнения итераций на ASM: " << clock() / 1000.0;

    std::cout << "\n";

    std::cin.get();

    system("pause");

    return 0;

}

**Код на С++:**

#include <iostream>

#include <string>

#include <algorithm>

#include <ctime>

using namespace std;

void RotateRight(string& s, const size\_t sz) {

    for (size\_t lsz = 0; lsz < sz; lsz++) {

        reverse(s.begin(), s.end());

        s.push\_back(s[0]);

        reverse(s.begin(), s.end());

        s.pop\_back();

    }

}

int main() {

    setlocale(0, "");

    system("color f0");

    std::cout << "Введите строку: ";

    std::string s;

    std::cin >> s;

    srand(time(0));

    RotateRight(s, 2);

    std::cout << "Сдвинутая строка: ";

    cout << s << endl;

    std::cout << "\nВремя выполнения итераций на C++: " << clock() / 1000.0;

    std::cout << "\n";

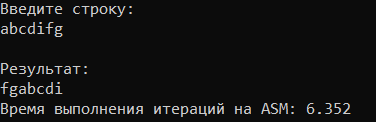
    std::cin.get();

    return 0;

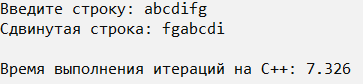
}

**Результаты тестирования:**

**На Ассемблере:**



**На С++:**



**Вывод:** в ходе данной лабораторной работы познакомились с языком Ассемблер. Написали вставку и сравнили время выполнения программы на С++ и на Ассемблере. После полученных результатов можно сделать вывод, что Ассемблер работает быстрее, но слишком сложный в реализации.