

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ  
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ  
«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ  
УНИВЕРСИТЕТ»

Кафедра ИИТ

Отчёт  
о лабораторной работе №1  
по дисциплине «Компьютерные системы и сети»

Тема: «Изучение аппаратно-программной архитектуры процессоров  
семейства Intel – разработка ассемблерной вставки»

Выполнил студент 2 курса  
группы ПО-11 Сымоник И.А.  
Номер зачетной книжки: 220220

Проверил: Савицкий Ю.В.

**Цель работы:** изучение программной модели микропроцессора, регистров, организации памяти; практическая работа с ассемблерными вставками

## Вариант 6

### Ход работы

**Задание:** Нечетные символы заменить на символ +.

#### Исходный код:

```
#include <iostream>
#include <string>
#include <chrono>
#include <Windows.h>
#include <string_view>

void AssemblyFunc(const char inputString[])
{
    std::chrono::steady_clock::time_point begin = std::chrono::steady_clock::now();
    __asm
    {
        mov esi, inputString
        mov ecx, 0

        replace:
        mov al, [esi + ecx]
        test al, al
        jz end_replace

        test ecx, 1
        jnz replace_char

        inc ecx
        jmp replace

        replace_char :
        mov[esi + ecx], '+'
        inc ecx
        jmp replace

        end_replace :
    }
    std::chrono::steady_clock::time_point end = std::chrono::steady_clock::now();

    std::cout << "Время работы алгоритма : "
        << std::chrono::duration_cast<std::chrono::seconds>(end - begin).count()
        << " секунд "
        << std::chrono::duration_cast<std::chrono::milliseconds>(end - begin).count()
        << " миллисекунд "
        << std::chrono::duration_cast<std::chrono::microseconds>(end - begin).count()
        << " микросекунд "
        << std::chrono::duration_cast<std::chrono::nanoseconds>(end - begin).count()
        << " наносекунд "
        << std::endl;
    std::cout << "Измененная строка: " << inputString << std::endl;
```

```

}

void Func(std::string& str)
{
    std::chrono::steady_clock::time_point begin = std::chrono::steady_clock::now();

    auto size = str.size();
    for (int i = 1; i < size; i+=2)
    {
        str[i] = '+';
    }
    std::chrono::steady_clock::time_point end = std::chrono::steady_clock::now();

    std::cout << "Время работы алгоритма : "
        << std::chrono::duration_cast<std::chrono::seconds>(end - begin).count()
        << " секунд "
        << std::chrono::duration_cast<std::chrono::milliseconds>(end - begin).count()
        << " миллисекунд "
        << std::chrono::duration_cast<std::chrono::microseconds>(end - begin).count()
        << " микросекунд "
        << std::chrono::duration_cast<std::chrono::nanoseconds>(end - begin).count()
        << " наносекунд "
        << std::endl;
    std::cout << "Измененная строка: " << str << std::endl;
}

int main()
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    std::string_view str =
"dsakdklas;kd;sajdlkjsadhjlsahdjkhksajkdhksjahdkjhaskhdjkhdkjhasdkjhasjkdjhkashdkhasdhjkashdkjassajdlkjsadhjlsjk
dhksjahdkjhasjkdjhkashdkhasdhjkashdkjksasdkdklas;kd;sajdlkjsadhjlsahdjkhksajkdhksjahdkjhaskhdjkhdkjhasjkdhksjah
dkjhasjkdjhkashdkhasdhjkashdkjassajdlkjsadhjlsahdjkhksajkdhksjahdkjhaskhdjkhdkjhasjkdhksjahahdjkhksajkdhksjahdk
jhaskhdjkhdkjhasjkdhksjahdkjhasjkdjhkashdkhasdhjkashdkjksadsadsaddawhc";

    std::string inputString = { str.data() };
    std::cout << "Оригинальная строка: " << inputString << std::endl;
    std::cout << std::endl;

    std::cout << "Функция с ассемблерной вставкой: " << std::endl;
    AssemblyFunc(inputString.c_str());
    inputString = { str.data() };
    std::cout << std::endl;

    std::cout << "Функция без ассемблерной вставки: " << std::endl;
    Func(inputString);

    return 0;
}

```

**Результат выполнения:**

Оригинальная строка: dsakdklas;kd;sajdlkjsadhjlsahdjkhksajkdhksjahdkjhaskhdkjhdkjhasdkjhasjkdhjkashdkhasdhjkashdjassajdlkjsadhjlsjkdhksjahdkjhasjkdhjkashdkhasdhjkashdjkasdsakdklas;kd;sajdlkjsadhjlsahdjkhksajkdhksjahdkjhaskhdkjhdkjhasjkdhksjahdkjhasjkdhjkashdkhasdhjkashdjassajdlkjsadhjlsahdjkhksajkdhksjahdkjhaskhdkjhdkjhasjkdhksjahahdjkhksajkdhksjahdkjhaskhdkjhdkjhasjkdhksjahdkjhasjkdhjkashdkhasdhjkashdjkasdsadsaddawhc

функция с ассемблерной вставкой:

Время работы алгоритма : 0 секунд 0 миллисекунд 0 микросекунд 500 наносекунд

Измененная строка: d+a+d+l+s+k+;+a+d+k+s+d+j+s+h+j+h+a+k+h+s+a+d+j+a+k+d+k+d+j+a+d+j+a+j+d+j+a+h+k+a+d+j+a+h+j+a+s+j+l+j+a+h+l+j+d+k+j+h+k+h+s+k+h+k+s+d+h+s+h+k+s+d+k+s+s+k+k+a+;+d+s+j+l+j+a+h+l+a+d+k+s+j+d+k+j+h+k+h+s+h+j+h+k+h+s+k+h+s+a+d+j+a+j+d+j+a+h+k+a+d+j+a+h+j+a+s+j+l+j+a+h+l+a+d+k+s+j+d+k+j+h+k+h+s+h+j+h+k+h+s+k+h+s+a+a+d+k+s+j+d+k+j+h+k+h+s+h+j+h+k+h+s+k+h+s+a+d+j+a+j+d+j+a+h+k+a+d+j+a+h+j+a+d+a+s+d+d+w+c

функция без ассемблерной вставки:

Время работы алгоритма : 0 секунд 0 миллисекунд 0 микросекунд 200 наносекунд

Измененная строка: d+a+d+l+s+k+;+a+d+k+s+d+j+s+h+j+h+a+k+h+s+a+d+j+a+k+d+k+d+j+a+d+j+a+j+d+j+a+h+k+a+d+j+a+h+j+a+s+j+l+j+a+h+l+j+d+k+j+h+k+h+s+k+h+k+s+d+h+s+h+k+s+d+k+s+s+k+k+a+;+d+s+j+l+j+a+h+l+a+d+k+s+j+d+k+j+h+k+h+s+h+j+h+k+h+s+k+h+s+a+d+j+a+j+d+j+a+h+k+a+d+j+a+h+j+a+s+j+l+j+a+h+l+a+d+k+s+j+d+k+j+h+k+h+s+h+j+h+k+h+s+k+h+s+a+a+d+k+s+j+d+k+j+h+k+h+s+h+j+h+k+h+s+k+h+s+a+d+j+a+j+d+j+a+h+k+a+d+j+a+h+j+a+d+a+s+d+d+w+c

Скорость выполнения функции без ассемблерной вставки выше. Это можно объяснить тем, что к ассемблерным вставкам не применяются оптимизации компилятора.

**Вывод:** изучили программную модель микропроцессора, регистров, организации памяти.