МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

“БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ”

**КАФЕДРА ИИТ**

ОТЧЁТ

по лабораторной работе №1

**«Программная модель микропроцессора. Ассемблерные вставки»**

Выполнил:

Студент 2 курса

группы ПО-9

Харитонович Захар Сергеевич

Проверил:

Савицкий Ю. В.

Брест 2023

**Вариант 1**

Задание – перевернуть строку.

Написать ассемблерную вставку, реализующую обработку строки согласно варианту, выданному преподавателем. Оформить ее в виде отдельной функции.

Реализовать данную обработку строки также в виде функции на С++. Сравнить быстродействие обоих вариантов.

**Текст программы**

#include <iostream>

#include <stdio.h>

#include <string.h>

#include <time.h>

char\* reverseStringC(char str[], const int size) {

char\* resStr = new char[size + 1];

for (int i = 0; i < size; i++) {

resStr[i] = str[size - i - 1];

}

resStr[size] = '\0';

return resStr;

}

char\* reverseStringASM(char str[], const int strSize) {

char\* resStr = new char[strSize + 1];

\_\_asm {

mov ecx, strSize

mov esi, str

mov edi, resStr

add edi, strSize

mov [edi], 0

dec edi

L1:

movs

sub edi, 2

loop L1

}

return resStr;

}

char\* generateStr(int size) {

char\* str = new char[size + 1];

for (int i = 0; i < size; i++) {

str[i] = (char)(rand() % 92 + 33);

}

str[size] = '\0';

return str;

}

int main() {

gets\_s(str);

std::cout << "ASM: " << reverseStringASM(str, strlen(str)) << std::endl;

std::cout << "C: " << reverseStringC(str, strlen(str)) << std::endl << std::endl;

std::cout << "ASM "; // проверка быстродействия ассемблерной вставки

char \* str1 = generateStr(256);

clock\_t time = clock();

for (int i = 0; i < 1000000; i++) {

reverseStringASM(str1, 256);

}

std::cout << clock() - time << " ms" << std::endl;

std::cout << "C "; // проверка быстродействия реализации на C

time = clock();

for (int i = 0; i < 1000000; i++) {

reverseStringC(str1, 256);

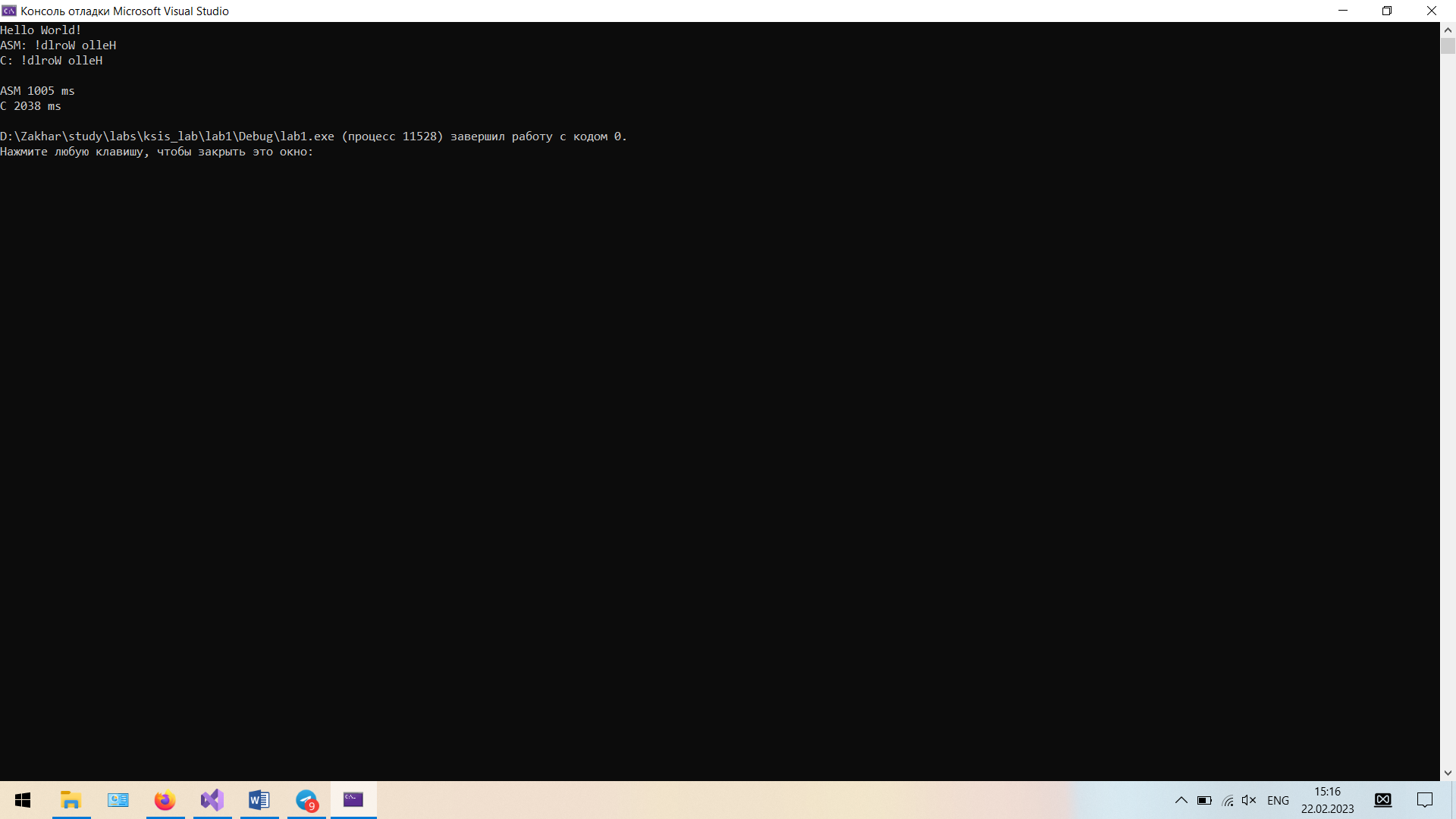
}

std::cout << clock() - time << " ms" << std::endl;

return 0;

}

**Результат работы**



**Вывод:** На больших объёмах данных функция, использующая ассемблерную вставку, показывает лучшее быстродействие, чем та же функция, написанная на языке C.